TECHNICAL REPORT
Software Hot Lines: A Preliminary Description
by
Brian T. Pentland
Revised March, 1990
CCS TR #104 SSWP #3140

CENTER FOR COORDINATION SCIENCE

Massachusetts Institute of Technology
Sloan School of Management
Cambridge, Massachusetts
Software Hot Lines: A Preliminary Description

by

Brian T. Pentland

Revised March, 1990

CCS TR #104 SSWP #3140

The author gratefully acknowledges the helpful comments from several of the participants in the research, as well from my advisors and friends here at the Sloan School of Management and the Center for Coordination Science.
Abstract

Technical support hot lines are an integral part of most complex software products. This paper uses data from a set of 13 software support hot lines to create a preliminary description and analysis of the basic process of providing technical support. Some key managerial issues are identified and the relationship between learning and coordination is discussed briefly. The results reported here are part of my on-going research into the role of organizational learning in coordination.
1.0 Introduction

Hot lines are everywhere. Whether you buy a can of soda or a computer system, the manufacturer generally provides a number to call in case you have questions, comments, or complaints. And while people seldom need help with their soft drinks, they do need help from time to time with their software and other complex technical products. Technical support is practically a requirement to do business in the computer industry: an "unsupported product" is hardly considered a "product" at all. As a result, hot lines (or some other vehicle for servicing customer inquiries) are commonplace features of software firms. But although the technical support function is commonplace (and less glamorous than activities like new product development), it raises some provocative problems for managers and organization theorists. It also provides an example of how an ostensibly simple task -- answering phone calls -- can give rise to a wide variety of coordination processes. The purpose of this paper is to create a preliminary description of the software support function that can be used as a basis for future research.

When customers need to know something, vendors try to respond. Each call that comes in is an occasion for a knowledge transfer. For this reason, technical support exemplifies what Machlup (1980) refers to as a "knowledge industry." With a typical software vendor, it is the function responsible for collecting, storing, distributing and in many cases actually creating the knowledge customers need. To some extent, this quality is shared by a wide range of service functions, from hospitals to travel agencies. But technical support exemplifies the particular problem of collecting and delivering
knowledge that is not readily available to any single individual, but is
distributed throughout the vendor organization. This distribution of knowledge
creates an interesting organizational problem: since no single person knows
everything, how can the organization respond effectively to customer inquiries?

The problem of how to organize the technical support function when
knowledge is distributed is the central theme of this paper, which is divided
into several sections. Section 2 describes the sample on which the paper is
based. Section 3 presents a brief case history of one of the firms in the sample.
This case includes many of the features other firms experienced in dealing with
growth in the size and complexity of the support task. Sections 4 and 5
describe some alternative ways hot lines are organized and how the process of
answering calls works in the organizations in the sample. Section 6 discusses
some of the practical problems of dealing with customers, human resources, and
linkages to the rest of the organization. The paper concludes with a brief
discussion of some implications for coordination theory.

2.0 Scope and Method of Research

Software vendors deliver technical assistance to their customers in a large
variety of ways. In this paper, I will refer to one of those ways -- the
telephone -- as "technical support." This label reflects common usage in the
industry, but it is worthwhile to step back briefly and consider the range of
services that firms use to provide practical knowledge and product information
to customers. These include documentation (printed and electronic), training
(classroom and computer based), and various kinds of specialized consulting.
Electronic bulletin boards and electronic mail are becoming increasingly common
means of interacting with customers, substituting for direct telephone contact in
Smaller, newer companies tend to have training, documentation, and technical support closely integrated, sometimes even in a single group. The grouping makes sense because the work is closely related and insufficient in volume to justify a separate staff for each function. Larger, older companies tend to differentiate the functions into separate groups, but usually maintain them as parts of a single customer or product services organization. Even when the functions are differentiated, there appear to be opportunities for cross-fertilization, as when trainers assist in writing documentation, or when documentation staff help with telephone support. So while this paper focuses on the activity of answering customer telephone inquiries, it should be seen as part of a broader range of activities that transfer practical knowledge to customers.

In order to create the following description of the technical support activity, individuals in 13 companies were interviewed. The firms in the sample were selected on an ad hoc basis from the Directory of Massachusetts High Technology Companies. In most cases, the manager of technical support was the sole interviewee; in some cases, other personnel were also interviewed. All respondents were assured of the anonymity of themselves and their companies. Interviews lasted about an hour and focused as much as possible on the operation of the telephone support operation. Pseudonyms for the companies included in the sample are listed in Figure 1 (p.40), along with the product area and some approximate estimates of the number of customers, products and platforms being supported, as well as the number of people providing the technical support service.
The sample reflects considerable diversity in terms of the products, hardware platforms, and customers. The products range from simple PC database products (PCCo) to extremely complex systems for automating entire multi-national manufacturing firms (BizCo), and range in price from a few hundred dollars to several hundred thousand dollars. The hardware platforms include everything from desktop PCs to engineering workstations to mainframes. The users of these products range from individual PC hobbyists to MIS managers to AI gurus. While many of the products are meant for use by developers (e.g., SmartCo, CompCo, ExecCo, CaseCo and others), some are sold directly to end users (e.g., PlanCo and PCCo).

3.0 A Case History

Although many readers may be familiar with calling a support hot line, they may be less familiar with working in or managing one. This brief case history is intended to provide some context for the discussion that follows. It describes the history of the customer support function at SmartCo, a company which sells expert system development tools. The description is based on a lengthy interview with a former manager of customer support who was also the first person hired as a full time support person. Although it is written in his voice, it is not a direct quotation. The case of SmartCo is particularly interesting because it describes the reorganization of the support function from one structure (personalized generalists) to another (de-personalized specialists) in the face of growing demand for increasingly complex services.
The Early Days at SmartCo

Initially, customer support, training, and documentation were all one group. We had a total of eight or ten people. People tended to be allocated half-time to two functions (like training and documentation), but everybody met together as a whole. At that time, this collection of activities was under the Director of Services. This grouping made sense because all three of these functions were concerned with teaching users about the product, helping them use the product, making the product accessible to them, and so on.

The company only had about two dozen customers at first, so there were between 6 and 10 customers per full-time support person. It worked out that you'd handle between 2 and 5 distinct customer contacts per day. For any given contact, there could be several actual phone calls, since you'd usually call back and forth rather than waiting on hold. It always seemed like you were taking a lot more than 5 contacts a day, but I remember looking over the daily call records (which we called blue sheets) to estimate that number, and it really never got much above that. Part of the reason is that you might spend a couple of days working on a single call.

The kinds of calls we got back then were often really deep questions. Our first customers were really AI pioneers. They were R&D-type places who were trying new things and wanted all kinds of special functionality. Often times, they wanted to do things that were not in the user level code, so we had to talk to someone in product engineering to find the appropriate functions for them to call. In a sense, customer support was filling in the gaps in the product for these customers. They knew they were getting a product which didn't do everything, but they still wanted to stretch it to fit their needs and our job was to help them do that.

The other kind of call we got a lot of -- the most frequent by far, actually -- had to do with installation problems. It really wasn't easy to install the software on certain machines, especially if the customer had an unusual hardware configuration. Many of these calls were routine, familiar problems having to do with not following the instructions, but some of them were real tricky, system level problems.

In the early days, everyone -- the whole company -- was in the same room with 6 foot dividers between the desks, so you could just call out your question and tap into the expertise of whoever was within earshot. But after the company moved into the new building (1986), people had separate offices (two persons per office), and you couldn't do that. If you knew who to ask, you might run down the hall and talk to them, but that wasn't necessarily very efficient or even possible.

One of the questions that management had at that time was how many customers a support person could really handle. This was important because they had to know what the staffing requirements would be as the customer base grew. Sales were starting to take off, so this was not an academic question. The support group met to discuss this issue and decided that 15 customers was a good number, and 20 was an absolute maximum. Beyond that, it would be impossible to provide the kind of personalized
service that the customers had grown to expect, and that the company wanted to provide.

Blue Sheets

Each call that came in was logged on a "blue sheet" which had various information, like the customer, the software versions, hardware platform, and most importantly, the description of the problem and the resolution. The blue sheets were really the life blood of support, providing not only records of call volume, but also history and continuity for the support group. We made 3 copies of each one. One copy we kept ourselves, one copy went into a file where all of them were collected, and a third copy went to the product support manager and the president.

In the early days, the president would actually look over all of the blue sheets, that's how important customer support was for the company. He put tremendous emphasis on being close to the customers. The customer service manager felt the same way, so much so that he used to distribute an illustrative chapter from *In Search of Excellence* to new employees to help make the point. Our job was to make the product work for them, not matter what. The support manager liked to hack code himself, and he encouraged creative solutions for problems and requests.

It was explicitly "no holds barred"; you had to handle everything. Sales reps in the field would tell the customers that we'd do anything for them, because part of what we were really selling was a relationship with the customer. So in addition to handling calls, we would pro-actively call the customers once a month, just to see how they were doing, if we could help, etc.

The relationship with customers was very personal. When they came for training, we'd make a point of having lunch with them, getting to know their application, and getting to know them. Customers had a direct line to their personal supporter, rather than having to go through the switchboard. We'd ask about their families, vacations, that kind of thing. At the annual customer meeting, we'd hang out and drink with our customers. There was definitely more than just lip service paid to the idea of getting close to the customers.

Of course, support people would occasionally be unavailable, due to vacations, illness, or other duties (more on these duties in a moment). To handle those situations, the concept of a "default support person" emerged. If you were out, the default person would take your calls. This was explained to customers, of course, who understood it as a necessary and desirable way of handling things when their regular support person was away. The default support people were generally very busy (this is putting it mildly), because they had all of their own customers plus whoever was away. The task was rotated among all support staff so that everyone did default once every 1-1/2 to 2 weeks. Although every effort was made to avoid the situation, it occasionally happened that only two people would be covering for the entire staff of ten supporters -- telephone hell. Nonetheless, the default support person provided a way to handle calls regardless of whether the regular support person was available.
The personal relationship with customers was important for the company for a variety of reasons. First, it meant that customers were more likely to succeed in creating productive applications. Early success stories were extremely important to the continued growth of the company, and support people were supposed to stay involved with their customers to help insure that success. We were also supposed to help the marketing people find and develop these stories into sales collateral: video tapes and brochures to hand out at trade shows. Another reason to stay close to the customer had to do with capturing innovative ideas and spotting areas where the product should be enhanced. Support people were supposed to be the conduit between the customer and product engineering for ideas, not just problems.

Growth, Growth, Growth

From 1985 to 1989, the size and complexity of the support task grew substantially in three different dimensions. First, the number of customers increased from a few dozen to several hundred. Second, the number of products that the company was selling grew from one to four. Third, the number of hardware platforms on which these products ran grew from two to around ten. What did not grow substantially was the number of support people. In August 1985, there were about 9 full time supporters. By August 1986, there were about 12, but that number stayed constant through the end of 1988. Although the maximum conceivable number of customers per supporter was thought to be about 20, we now had between 40 and 50. Not all of these customers were actively developing applications, but the number was still way too high.

Part of the reason support staff was not increased was that the company was not quite breaking even on its sales. There were some layoffs, and new slots could be justified only if they would be profitable. For a variety of reasons, it was very hard to show that adding another support person would increase profits, so the existing staff was forced to take on more and more customers. Since support, training and documentation were all under the manager of customer services, he would sometimes take an open slot away from documentation and give it to support, but that was rare.

The large and growing number of customers put support in a very awkward position. We claimed to be providing personalized service at a very high standard. Over the years, we had developed a great deal of pride and esprit de corps around this theme. But now it seemed hypocritical to talk about a level of service we could not realistically provide. We could not proactively call customers; in some cases, we couldn't respond to calls that weren't "crisis level" for the caller for several days, if at all. When things grew to this point, it was really just fire-fighting.

Other Responsibilities

As I implied above, support people did more than just answer the
phone. Their other main functions included quality assurance, marketing, consulting, and providing input to development on what customers wanted. Of these many roles, the quality assurance role was among the most time consuming. When a new product release was being worked on, it was given to customer support for an early shakedown. Support people would report bugs to product engineering and in the process, familiarize themselves with the new release that their customers would soon be using. This role -- bug reporting -- was part of the regular duties of a supporter, of course, since the process of responding to customer inquiries often involved diagnosing and reproducing problems.

But quality assurance for a pre-release product was a special responsibility because customer support had to sign off on any product before it could be shipped. The theory was that if we were going to support it, we should be allowed to say if it's ready. This formal veto power was a great source of tension at times. A sales person would say, "I can close the sale now if I have those new features"; product engineering would say, "the new features are ready"; all the top brass would say, "we really need the revenue"; but the support person would say, "I haven't tested it yet" or "It's still buggy." Testing was known to be important, but it was easy to lose sight of that in the rush to claim revenue. Support held its ground in most cases, although it meant pulling people away from regular support assignments to do the testing. When a whole new product was coming out, being the default support person could be a nightmare, because so many other people would be doing testing.

The other responsibilities were less critical, but still took time. Marketing activities consisted mainly of getting good stories together for the sales people, as I mentioned earlier. Consulting was different, kind of an extension of personalized support to the extreme. You might work personally with a customer, one on one, for several days at a time. This was a lot of fun and it was also good for the career development of the support person. These were considered real plum assignments; very desirable indeed. But they also took you away from the phone and added to the work of the default person.

Implicit Specialization

Because supporters were assigned to customers, we were supposed to be generalists. Customers use all aspects of the product, so supporters need to be familiar with all aspects of the product. But in practice, there was always some degree of specialization that took place.

The first kind of specialization was by hardware platform. As I mentioned, many of the support calls were related to installation, and these problems were all hardware specific. Also, the "native language" of the two main hardware platforms was different in the early days, and the product itself has some minor differences, as well. All of these factors contributed to a tendency for certain supporters to know more about one kind of hardware than the other.

The second kind of specialization was by software functionality. The product has several major pieces that are logically distinct. Each piece
has a different purpose, calls different functions, and has different idioms for use. For example, the user interface involves an large number of features for creating special windows, menus, graphics, and so on. When customers make heavy use of interface features and stretch the limits of the product, their support people will tend to develop some expertise in the interface. The same goes for any other part of the system.

The third kind of specialization that developed under the personalized service system was by product. Initially, the company had only one product, so this dimension was nonexistent until mid-1986. But when the new products did start coming out, certain support people were specifically assigned the job of supporting the new customers. As the customer base for that product grew, additional supporters were brought into the pool.

The tendency to specialize along each of these three dimensions was reinforced by the quality assurance work. When something needed testing it was given to the people who knew the most about the platform, feature, or product in question. In the process of testing, the support people would attempt to exercise every aspect of the system and become quite familiar in the process. So although the formal assignment of customers to supporters implied that supporters were first and foremost generalists, the level of specialization among supporters was constantly increasing.

Knowledge Overload

As mentioned above, the growth being experienced involved more than just more customers. Existing product had additional features which in some cases involved rather subtle concepts of knowledge representation and reasoning. The marketing strategy for the company involved supporting new hardware platforms, but that meant learning about new operating systems (e.g., UNIX, VMS and even DOS), new file systems, new machine configurations, and so on. New products added entirely new sets of functionality and potential problems to the support person’s list. Although not all products were available on all machines, and product were remarkably standard across platforms (a real tribute to the product engineers), there still tended to be a multiplicative effect in terms of knowing about each product on each different machine. In this situation, it was not uncommon for the customer to know more about their hardware and operating system than the support person, which was embarrassing and damaging to the credibility of the company, even if the person was an expert in the company’s own products.

One response to this knowledge overload was to create mechanisms for sharing expertise among supporters. So we created the Red Book, which was a 3-ring binder that each support person used to keep track of solutions to typical problems. It included things like how to make a certain kind of menu, how to install the software on a network with an unusual kind of file server, things like that. When people solved an interesting problem, they would write up a page about it, including any code they had written, and circulate it to the group. People would add it to the appropriate section of their personal Red Book and everyone was kept up to date.
Another mechanism was the SOS system, which we implemented on our VAX. It was essentially an electronic version of the Red Book, with sub-directories for each topic area. You could log into the SOS account and browse through various messages about past problems and their solutions. Although this was better than the Red Book in some respects (it was centralized), it wasn't really that great. Messages were stored in separate files which were not very convenient to use. This primitive online system was made available to users, as well, on a micro-VAX housed at the company headquarters. Apparently, users were glad to have this resources, but were disappointed at its performance and accessibility.

The New System

The support function at SmartCo has recently been reorganized. In the new system, customers do not have personal support people. Rather, they call in to a dispatcher who verifies that they are entitled to support and tries to identify what kind support they need. The dispatcher then routes the call to a support person who specializes in that area. This system formalizes and extends the kinds of specialization that were implicit under the earlier organization. This allows support people to focus on the portion of the product they know best. It also cuts down on the number of times that the customer discovers that they know more than the person they are relying on for support.

The old "blue sheets" have been replaced with a special purpose database, developed by our database vendor for their own customer support, and modified to suit the details of the new setting. Each call initiates a transaction in the database which remains "active" until it gets closed out by someone resolving the problem. Supporters can browse their active calls, and managers can get reports on the overall state of the support effort. All entries are time-stamped, so it's possible to track exactly how long each step in the process takes.

The second innovation in the structure of customer support is to move to a "two-tiered" system. The lowest level of support is purely "online" which is now delivered through CompuServe, so it's faster, more available, and more usable. People call in and browse through various topic areas populated with all kinds of questions and answers. The second tier is the "dispatcher-specialist" service just described. This means that some of the calls that would otherwise be handled by customer support people are handled by CompuServe.

In the new system, each person has 4 or 5 specialties. In some sense, we are still generalists, but in a much more limited sense than before. There is a notion of a "primary" and "secondary" specialist, as well. Any given person will be the primary specialist on one or two areas, and a secondary specialist on the others. Specialties arise from the kinds of hardware or software discussed earlier. Each specialty area has at least one person covering it, and as many as three.

This is the history of technical support at one organization, but it
demonstrates some of the issues involved in managing that function: the demands on the staff, the hectic pace of work, and the complexity involved in something as apparently simple as answering the phone. As I proceed with the analysis, I will draw on this case and on the other firms in the sample to describe a variety of organizational forms and management issues as observed at SmartCo and elsewhere.

4.0 Alternative Organizational Forms

The support function as SmartCo went from being a group of personalized generalists to a highly structured team of specialists. These are only two of the possible ways of organizing the technical support function. Figure 1 indicates the organizational form used in each of the other organizations sampled. The three main systems observed are (1) "specialists," where an operator takes calls and routes them to the appropriate specialist; (2) "generalists" (with or without a dispatcher); and (3) "personal assignments," where each support person has a specific list of clients they support. Each of these forms will be discussed in the following section.

4.1 Specialists

The specialist system appeared in five of the organizations visited. The differentiating feature of this system is that there are formal assignments of responsibility for different kinds of knowledge. Knowledge is divided in several ways. As mentioned earlier, the main divisions are organized by hardware platform and product or product feature. In those companies that sell products on multiple platforms, there are almost always specialists for each major platform (e.g., IBM and DEC, or UNIX, DOS, and VMS). Even though the vendor's own product may be nearly identical on each of these platforms,
customers want to talk to someone who "speaks their language." There may also
be substantive differences in how a product works between environments that
require special knowledge. One support manager explained it as follows:

When customers call, they have a specific operating system in mind. We run on both DEC and IBM, and an IBM customer doesn't like to
talk to a DEC support person or vice versa. They just have completely different languages. All of our people know our product
inside and out, but they may not be very experienced in the various operating systems. That's why I need to have people with an
experience base in each of the major operating systems we run under.

For companies with large products (with many separate sub-systems) or
companies with multiple products, there is often a division of labor by product
or product area. At SmartCo, there are support specialists for each of the
main sub-systems within the product. This suggests that there is just too much material for the whole staff to become fully versed in at a level adequate to respond to user questions. At SpreadCo, the main specialization was by hardware platform, but a new specialty had emerged to support the company's database connections. This was an area where an experienced person could respond to user questions "ten times faster" than the other staff, so specialization made sense.

There is also some division of labor within support groups between
"application" or "user level" questions versus "technical" questions. This is a much more subtle distinction which depends on classifying what kind of information the user needs, rather than what kind of machine they own or what product feature they are using. ExecCo has specialists who handle application questions (how to use the product to achieve a certain result), while most of the support group are dedicated to providing "statement level" technical support (how to use a specific product feature correctly). Since these two areas are
closely interrelated, it may be difficult in practice to make the distinction in all cases.

4.2 Generalists

Generalists were used to provide technical support in five of the companies visited. In this system, the staff are responsible for any question on any product, and customers are not assigned to any specific supporter. This system appears to occur mostly in firms where there is only one hardware platform and one product, or where the staff is too small to allow much specialization. In terms of the number of products and platforms they support, these groups correspond to the "specialists" in other firms, but they appear to be generalists because they cover the entire range of questions that come up. Two of the firms with generalists (CompCo and OffCo) actually support a large number of hardware platforms, but they have only two full-time support staff, so specialization is unfeasible.¹ CaseCo has a large staff with three levels of seniority, but since they sell a single product on a single platform, all of these people are generalists.

4.3 Personal Assignment

Assignment of clients to specific support people appears to be driven by two factors: installation specific details and the desire to maintain close customer relations. For example, BankCo uses this system because the installation and operation of their software is highly customized to each user site. A supporter who was unfamiliar with conditions at the client site would

¹ In fact, CompCo does have specialization within its technical support group, but it occurs between the groups in the US and in Europe. Since communication between these groups is limited, the two US support staff are forced to fend for themselves.
waste a great deal of time in getting an explanation of all the relevant details for each call. Personal support people provide continuity and familiarity with unique circumstances at the customer site. The also allow the vendor to keep tabs on whether the customer is satisfied, whether other business opportunities exist with the customer, and so on.

4.4 Hybrid systems

There are a variety of hybrid system that are difficult to label concisely. SaleCo uses a system of personal assignment of clients to support people, but rather than assigning clients to individuals, they are assigned to teams of specialists. Each team consists of an operating system specialist, a communications specialist, and a product specialist. These specializations correspond to the needs of a typical client. Although this system requires additional staff to operate, it combines the continuity of personalized service with the depth of skills provided by specialists.

BizCo combines specialists and personal assignment in a different way. In their system, users have two sources of support: a hot line staffed by technical specialists and a local representative who is assigned to the client and can provide on-site assistance. Many firms have consulting services which can be called in when ordinary support is inadequate. What differentiates BizCo is that the local representatives are also the people who assisted in the design and installation of the system. Because they are assigned to particular clients, these consultants provide continuity.

In addition to these formal hybrid systems, there are a number of informal hybrids that develop. The most common of these is informal specialization,
which was observed at nearly every company in the sample. As was the case at SmartCo, informal specialists seem to emerge quickly within support groups. When a customer calls with a challenging problem, the person who takes that call becomes a "local expert." When the next call comes in on a related topic, there is a natural tendency to "ask the expert." Before long, such people begin to get all of the difficult calls on their informal "specialty," and their expertise grows quickly and is reinforced by formal assignments (such as participating in design meetings for a new release or quality assurance).

Another common kind of hybrid develops when customers ask for a particular support person (even though there is no formal assignment). Some companies discourage these requests because they interfere with the formal call assignment process. When they are honored, requests from callers create a kind of informal personal assignment system on top of whatever other system is formally in place.

5.0 The Call-Response Process

Given the diversity of this sample, there is a remarkable degree of similarity in the basic process of answering customer calls. In this section I will describe the call-response process using concrete examples from specific firms to illustrate the details. In addition to tracing the process, I will discuss some of the implications of how each step is implemented.

5.1 Classifying the Call

The classification of the call is the first step in determining the appropriate response. When a call comes in, it usually goes directly to a "hot line support" number. Barring wrong numbers, all calls to this number are
seeking technical support. Some smaller firms don't have a separate technical support number, so the regular switchboard operator or receptionist serves a preliminary screening function. At this preliminary level, the caller only needs to identify him or herself as seeking technical support.

Given that a caller wants technical support, the interaction proceeds by checking whether the caller is entitled to that support. Entitlement is usually based on having a valid license for the software and an up-to-date maintenance agreement. Maintenance agreements are required for almost all products where technical support is offered, and typically cost 15% of the initial purchase price per year. Maintenance accounts for as much as 50% of total revenue in some of the companies sampled, so entitlement is an important issue.

Conceptually, checking the entitlement of the caller is part of the overall process of classifying the call. A taxonomy of hot line calls could start by distinguishing between two kinds of customers: those with valid maintenance agreements and those without. Those without agreements can be further subdivided into those with valid licenses who have let their maintenance lapse, and those with pirated copies who are trying to steal services as well as software. Among the people with lapsed maintenance, the lapse may or may not be intentional, and so on. Each of these distinctions can gave implications for how the person taking the call may choose to respond. For example, some companies will provide technical support to customers whose maintenance agreement has expired, and then use the fact that support has been given as a tool to help get the client to renew.

Although classification is logically the first step in handling a call, every
call cannot be neatly classified in advance. Even when an entitled caller states explicitly what they want, further discussion may reveal that something else is actually the problem, or that there are multiple problems, or that there is no problem at all. The classification of a call remains problematic until the call is actually resolved, because one can never be sure that a problem has been correctly identified until it is actually solved. This poses a further problem because in many cases, there is no feedback to the hot line concerning the actual outcome. Calls are treated as resolved when the caller stops calling, whether the underlying problem was resolved or not.

Likewise, the issue of entitlement to support is continuously subject to negotiation. Several respondents mentioned the problem of customers asking for services which went beyond what should properly be covered by their maintenance agreements. For example, customers may want you to debug their program for them, or teach them how to write a program in the first place. What starts out as a reasonable request from an entitled customer can turn into an unreasonable request quite easily.

5.2 "Logging" the calls

Nearly all of the organizations surveyed keep records of the calls they receive. In most, calls are logged into an electronic database of some kind. Some of the firms (including OffCo, CompCo, and until recently, SmartCo) keep track of their calls on paper only. In either case, the information kept in the records is similar from firm to firm: identifying information about the caller (name, phone, license number), the product and version number, the hardware platform and operating system, and frequently, the classification of the call.
Call logging serves a variety of purposes. First of all, it allows the support staff to track the history of a particular problem over the course of several actual phone conversations, which may extend over several days. Since not all problems are resolved on the first contact, records of calls are an important means of insuring continuity. The call log also provides a means of "covering your ass" when a customer complains that they aren't getting proper attention. If the call log reveals that in the last interaction the customer agreed to try something and call back, but no record of a return call, then such a complaint would effectively be diffused.

Second, the records provide an indication of how many calls are coming in, on what subjects, from which customers, and so on. Data on the volume of calls at various times of day is used to determine staffing requirements in some of the organizations visited. The subject matter of calls provides information about specialties that may be in demand or problems that may be developing. It can also reveal product areas that need improvement, or opportunities for new features to address new customer needs. Data on who is calling can reveal customers who are discontented or need additional training. The companies with electronic call databases could generate these kinds of reports (and others) much more easily, and believed this information was useful. The extent to which these reports were actually used was not clear from the interviews.

Third, these records can provide a history of problems and solutions. If used effectively, solutions to prior problems can be a helpful guide to solving current ones. Several firms with electronic call databases (including CaseCo and ExecCo) have systems for indexing their calls to make them more accessible to the staff. In some cases, the calls are indexed by their classification; in
others, they use the actual text of the problem description (and solution) to create the index. Firms with paper call logs also filed them in indexed files, but they seemed less able to use the previous call sheets for answering current calls.

Several of the firms I visited also maintained databases on problems and fixes, separate from the actual call history. The purpose of these systems is to organize and store useful information on certain topic areas. At SpreadCo, support staff were asked to write up a discussion of any problem that took more than 30 minutes to solve. These discussions were initially kept on paper, but in recent years have been indexed in an electronic database for use by the entire group. SmartCo has a system with a similar purpose and history. The early, paper-based version of this system at SmartCo was kept in red notebooks. Each support person was responsible for updating his or her own book. The goal was to create a centralized resource of practical information about how to use the product that could ease the burden on the support staff. Towards this end, the system at SmartCo migrated to electronic mail and then to a bulletin board that was made accessible to customers.

5.3 Assigning the Call: Knowing who Knows

Each call has to be assigned to a support person for response. There were a variety of systems of accomplishing this in the sample, as shown in Figure 1. The simplest process was used in firms where customers were pre-assigned to specific support people. If their support person was in the office to take the call, then the assignment problem was solved. However, people take vacations, attend meetings, and have other assignments from time to time. This creates the need for some kind of backup system for the assigned person.
In SmartCo (while this system was in use), there was one "default support person" on duty every day out of a staff of 12. Although the system worked well when only one or two people were out, it quickly became "telephone hell" when the group was short-staffed.

The next simplest assignment process was used in firms without specialists. The limiting case was SmallCo, where all calls were handled by a single person. A more representative system was at CompCo, where the operator took messages for technical support on the standard pink "While you were out..." message slips and put them in a designated "in-box." The support people would pick messages out of the box and return the calls, making sure not to leave any messages there for too long. In this kind of system, people take calls as soon as they are free to do so. Since the support people can sort through the messages and decide which ones to take, they have the opportunity to self-select the calls they would rather handle. Although no detailed data was collected on how these decisions are made, the key point is that calls are not just handled on a first-in, first-out basis. In principle, the staff has the opportunity to select calls which may be easier, more familiar, more challenging, or more fun. While this creates the possibility of shirking, it also allows the staff to pick the calls they are best able to answer efficiently.

The process used in the "dispatcher/specialist" system is somewhat more complex. In this system, the assignment proceeds from the classification of the call. An "IBM call" would get assigned to the "IBM group," and so on. Within the specialist group (which might consist of several support people), calls may be assigned based on who is available or there may be further specialization. Although this layering could proceed indefinitely, no firm in the sample had
more than two layers. Another minor complication in this process is that if a call is mis-classified, it must be reassigned.

5.4 Action at a distance: diagnosing the problem

Once a call has been assigned to someone, the next step is to ascertain the problem. In many cases, the customer has a specific question that can be answered immediately and easily. Support managers say that "easy calls" are common in the technical support organizations sampled, comprising anywhere from 50% to 80% of the total volume of calls. "Easy" questions often involve documented features that the user could have read about in the manual, or routine installation problems. One of the hazards of life on a hot line is the boredom that arises from answering dozens of trivial calls, explaining the same simple procedure over and over again.

But not all calls have immediately obvious answers. In many cases, the caller reports symptoms that are insufficient to diagnose the problem. In these cases, there are several paths of action available, each of which requires some form of repeated or prolonged interaction with the caller. The most common case is the call-back, which can work in two ways. If sufficient information is provided in the initial contact but the answer is unknown, the support person may need to do some research. This research can involve consulting manuals, records of prior calls, or other support people. It can also involve attempts to replicate the problem on a computer at the vendor site. When this initial research is complete, the support person calls back either with an answer, or if the problem is still unsolved, a request for additional information.

When the initial information provided by the caller seems inadequate, the
support person will usually ask the caller to collect more. If the caller is relatively sophisticated, the support person can issue rather abbreviated instructions (e.g., "See if anything looks funny in the system file and get back to me.") But if the customer is relatively naive, the research process gets more complicated. A common occurrence in technical support interactions is what I call "phone coaching," where the support person gives keystroke level instructions over the phone to a caller who is otherwise unable to perform or interpret the necessary diagnostic tests: "Type in D - I - R - space - A - colon and then hit return. OK? Now tell me what it says..." Phone coaching requires incredible patience from both parties, because it is extremely tedious and often very frustrating.

Phone coaching is much more common in organizations that support what Rockart and Flannery (1983) call "non-programming end users." OffCo, for example, sells office automation software that runs on the UNIX operating system. While users have been trained in the use of OffCo's products, they are quite often completely ignorant about how to use UNIX or the system editor, EMACS. Even the simplest diagnostic tasks require the use of these tools, so phone coaching is relatively common in the support of this product. The users of CompCo's compilers, on the other hand, are trained software engineers and analysts. Keystroke level phone coaching is far less common for these kinds of users.

Even where users are quite sophisticated, the support person still confronts the basic problem of getting information about what is happening on their system. The general pattern of coordinated action at a distance cannot be avoided unless the support person accesses the caller's computer directly.
Dial-in support is increasingly common among software vendors and is becoming standard practice among hardware vendors. FastCo, for example, requires dial-in access as a condition of their maintenance agreement. OffCo, CaseCo, BizCo and others also use dial-in capability as part of their diagnostic bag of tricks (although some customers refuse to allow remote access due to security restrictions).

While dialing-in may facilitate gathering information to diagnose the problem, it can complicate the issue of who will actually solve the problem. As mentioned above, there is an inherent ambiguity over the limits of responsibility the vendor has for solving customer problems. Several support managers noted that some customers want you to do their work for them, to debug thousands of lines of code, to optimize an algorithm, to design a user interface, and so on. At CaseCo, the support manager noted that once a support person dials in, they are hooked: "There's a kind of unwritten rule that once we look at the code, we have to fix it." Although this rule has limits, it is easy to see that a customer could be unhappy with a support person who edits their broken code, verifies that it's broken, and then leaves it that way. Conversely, they could be just as unhappy about a support person making changes to code without asking or explaining what was done. So although dialing in facilitates the diagnosis of problems in some cases, it does not necessarily aid in their solution.

5.5 Finding the Answer vs. Creating the Answer

As noted in the discussion of problem diagnosis, not all problems have immediate answers. This assertion leaves open the question of whether the sought-after answers are "out there" somewhere, waiting to be found, or whether they need to be created anew. The answer is that both situations occur
routinely in software support. The support people I interviewed often spoke in terms of "finding an answer", but they also spoke in terms of "developing a workaround." The distinction is clear in some cases, but rather subtle in others.

The clearest cases of "finding" answers arise when a support person gets a call that can be answered by looking at documentation or a prior call sheet. There are a variety of resources that support people can consult to find answers. Users of PlanCo's project management software require special plotters to create hardcopy of their projects' critical path charts. Since there are a large number of plotters available, each requiring appropriate configuration to work correctly with the software, PlanCo's support staff keeps files on how to configure each plotter. When a customer calls with an output problem, these files provide the resource needed to find the answers.

Another way of finding answers is to ask other people. As noted in the introduction, no single individual knows every aspect of a complex software product. Therefore, finding answers often requires interaction. In a few of the firms sampled (including BankCo, FastCo, and CaseCo), support people who are answering phones work face-to-face in a single room. When a difficult question arises, they can just call out to the others. SmartCo used to house support and product engineering in a single room with low dividers, so the entire development staff was within earshot if a tough question came up. But most companies in the sample house their support people in separate offices, so the immediacy of face-to-face interaction is lost. It is common practice to "run down the hall" to find someone who may know an answer. When the person is found, the answer is found.
But not all answers can be found so easily. In fact, some of the calls that come in pose problems which are completely new and have no ready answers. A typical example of a "new" problem arises when a customer upgrades some part of their system (hardware or software) that is sold by a different vendor. The example mentioned by one interviewee was an operating system. The new version of the operating system may not be fully compatible with the product, a fact which was previously unknown to the vendor's support staff. It turns out that customers often discover such incompatibilities before vendors do, sometimes because they are able to get new versions in advance of the vendor, or because they have a specialized configuration that the vendor would not otherwise have occasion to test.

In situations where incompatibility is the problem, diagnosis may be straightforward but solutions are not. Two courses of action are possible. Either the customer can go back to the earlier version that worked, or the vendor can quickly try to work out the problem and change their product to work in the new operating environment. In the short term, the only feasible option for the customer is going back to a previous version. But the longer term solution, changing the software, clearly involves a creative process of adapting the code.

There are other kinds of situations that require creating new answers. A variation on the compatibility problem just discussed can arise when a customer attempts a new application. New applications can create new problems, even if everything is ostensibly working correctly. At SmartCo, customers were constantly attempting to create innovative applications, and part of the role of
customer support was to fill in where the product left off. A typical example of this was in creating interfaces for sharing data with other products or between different kinds of machines.

Another situation where new problems arise is when a previously unreported bug is discovered. The diagnosis process involves the same kind of steps mentioned above, but the outcome is that the vendor's product does not work correctly. It is worth noting that while many callers complain that the product is not working as expected, not all of these call represent bugs. Part of the diagnosis process is to ascertain if the customer's expectations were correct, or if they misinterpreted or perhaps never even opened the manual, and so on. In any case, the support person has to figure out what the program written by the customer should do. In many cases, the expected performance is unambiguous, but that is not always true. The support manager at CompCo noted that the "pinnacle of the support job" was what he called "lawyering": deciding on what particular statements in the language should mean in certain contexts.

The initial response to a new bug is generally a workaround. If there are multiple ways to achieve a given result, then an alternate method will be researched, tested, and offered to the caller as a temporary fix. Known bugs have known workarounds, but new bugs require creative effort to fashion a new, acceptable solution to the problem. According to the support managers I interviewed, customers generally accept the necessity of workarounds for problems until a permanent solution can be obtained. The important thing from the customer viewpoint is that they be able to continue doing their work. The support managers I interviewed said they were keenly aware of this objective,
and generally adopted it as their own.

5.6 Bugs and Enhancements: Prioritization and Tracking

As new bug are identified, the support staff enter them into a system for prioritizing, tracking, and fixing them. The general outline of this system was remarkably similar across the entire sample. For example, every manager that mentioned their system for prioritizing bugs used basically the same categories. The dimensions of the categorization were severity (does it prevent work?), impact or prevalence (for whom, in what situations?), and avoidability (do we have or need a workaround?). The most urgent category of bugs prevented all work in all situations (no workaround available), followed by bugs that prevented work in some situations (and had workarounds), followed by bugs that were merely nuisances and could be avoided by simply not going out of your way to demonstrate them (no workaround needed). The least urgent category in every case was "enhancement requests," features that work correctly now but that somebody wants to work differently. Some firms had additional categories, but the basic outline was similar everywhere.

Each firm has some kind of release cycle for updating its software. The maintenance agreement allows customers to get these updates as they are produced. Some firms were very rigid about only updating the product on the official schedule, while others were more willing to create special "patch tapes" that could be sent to customers as needed. This seemed to be a function of the newness and complexity of the software, as well as the kind of relationship the vendor wanted to maintain. In any event, exactly what gets included in a given update or patch tape is subject to negotiation. In most cases, the manager of technical support has input into the prioritization and selection
process. In some cases, as at SmartCo, the support function exercises veto power over what goes out.

6.0 Managerial Issues in Technical Support

Technical support involves a variety of considerations that shape the priorities and possibilities for how the function is organized and managed. Some of these issues are implicit in the description of how calls are handled. The purpose of this section is to identify these issues explicitly.

6.1 Customer relations

One of the key issues in managing the technical support function is deciding what kind of relationship you want to maintain with your customers. Since technical support is the primary point of on-going contact between a vendor and its customers, it is a focal point for managing the relationship. The kinds of relationships that are possible are partly determined by the nature of the product (simple versus complex, cheap versus expensive), but it is also a strategic decision because good technical support can be used as a differentiating feature for a company in a competitive market.

Customer relations in the firms I visited ranged from rather impersonal and distant to extremely close and attentive. The more distant relationships were associated with the higher volume, lower cost products. As one support manager pointed out, it just isn't possible to provide personalized service when the product only costs a few hundred dollars. The more expensive the product, however, the more service is expected for the 15% maintenance fee. Likewise, if the vendor has relatively few customers, then service and support are important ways to keep existing customers happy and develop good references.
for subsequent sales.

An example of a basic policy decision regarding the quality of support is whether or not to commit to a specific response time, and what that time should be. Companies in the sample ranged from "sometime that day, or the next day at the latest", to "absolutely within four hours", to "one hour at the most." In some companies, the level of response was managed down to the details of who should pick up the phone after how many rings, and various forwarding/hunting schemes were employed in the local phone networks. It was not clear from the interviews how response time objectives were actually measured or what the consequences of failing to meet the objectives would be.

The companies with the closest relationships seemed to be the ones with very few (but very large) customers. Given their small size, BankCo and SaleCo had extremely elaborate systems for installation support, training, technical support, and product customization. The support manager at each firm emphasized that their job was to provide the best possible service to these clients. Although BizCo has a relatively large customer base, it also has a very sophisticated set of services that it offers to clients. The fact that these firms refer to the licensees of their software as "clients" and not "customers" reflects the relatively service intensive nature of their business. Vendors like CompCo or PCCo ship the software and the manuals and more or less expect people to figure it out themselves. Support is still an essential part of doing business, but it occupies a lower profile within the firm.

6.2 Human Resource Issues

Staffing the technical support function poses a fairly uniform set of
problems in the organizations I visited. The basic problem that managers articulated is finding good people and keeping them. The underlying problem is a kind of catch-22 for the support manager: people who have the skills to do the job generally don't want to, and people who don't have the skills require many months of training before they become really useful. Several of the managers I interviewed offered the opinion that 6-9 months of training is required before someone can handle calls competently on their own. Once they are trained, they no longer want the job. Part of the reason for this skills bind is that technical support is seen as an unglamorous function. As one manager explained, "it's always a stepping stone to somewhere else, it's never an end in itself."

But an even bigger problem in managing a technical support group is "burnout." Several factors make hot line support staff prone to burnout. First of all, almost every caller has a problem. In many cases, they have been working on it for quite some time before they call, so by the time they do, they are tense, angry, and looking for a scapegoat. When a problem is fixed, that's just what's expected; praise and positive feedback from customers is scarce. As a result, support staff get yelled at all day. Some of the interviewees claimed to be able to take indefinite amounts of abuse without it bothering them, but they acknowledged that it gets to most people after a while. Like many other kinds of service, hot line work is emotional work (Hochschild, 1983).

The other factor contributing to burnout is the fact that most calls to support hotlines are quite routine. As mentioned earlier, support people spend a great deal of time answering trivial questions, repetitively explaining
installation procedures, and so on. After amassing a couple of years of experience in diagnosing tricky systems errors, a senior support person rarely wants to explain to someone how to use an editor or load a tape. They have better things to do, and when they decide to go do them, it usually involves moving out of technical support.

Several of the firms visited have innovative systems for limiting burnout and providing growth opportunities to technical support staff. For example, CaseCo uses a two-tiered system where "primary" support people talk on the phone with customers and handle all of the routine problems. "Secondary" support people are used as backup for the "primary" people and generally don't talk to customers unless they need to. As a result, they get more challenging work and are spared the boredom of explaining trivial things to novices. SpreadCo divides its support group into three sub-groups, each of which has a "technical group leader." This position entails additional status and responsibilities and provides a growth path for people in support.

Several of the firms (including SpreadCo and CaseCo) also limit the amount of time that support people actually spend taking calls. In firms where this practice is used, half-time (4 hours a day) seems to be the norm. The time spent away from the phones is devoted to researching outstanding calls and special assignments like quality assurance or consulting. These other activities give the support person a break from the emotional stress of listening to angry customers while also providing a chance to learn new skills.

6.3 Relationships within the organization

Technical support is generally not an isolated part of a software company,
but the particular ways in which it relates to other functions represents an important set of managerial issues. In the organizations visited, technical support had a variety of different relationships with other parts of the organization. In general, technical support gathers information that is potentially useful to other parts of the organization. This section outlines some ways this information is sometimes used.

Product development is an area that is potentially very interested in the information collected by customer support. Every support group I visited played some role in reporting and prioritizing bugs and enhancements. Customers report bugs through the hot line, so this group is in a position to know which ones are relatively more important and urgent. Quality assurance and testing is another role that technical support participated in. In some firms, there was a separate quality assurance group, but others firms used technical support people for this purpose. The job of testing a product before release fits well with the bug reporting and prioritization function. It also gives the support staff a chance to learn about a new product before customers get it.

The natural continuation of the quality assurance function is the management of the beta test process. "Beta test" refers to the common practice of giving advance copies of new software to selected customers for actual field testing. Customers who participate in beta test typically receive a discount, free copies, or some other special consideration. Involving technical support in beta testing makes sense for a variety of reasons. First of all, technical support is oriented towards identifying and reporting bugs, which is the objective of the process. Also, technical support knows which customers have requested features that are present in the new release and therefore likely
to exercise them during the test period. Finally, allowing the support group to manage the testing gives them a chance to develop experience with the new release before the entire customer base gets it.

The manager at SpreadCo provided some interesting reasons why technical support is well situated to orchestrate the beta test of a new release. When product development ran beta tests, only 10% of the customer given the new release would actually install it, and of those, only a few would use it. As a result, very little testing actually went on. When technical support started running beta tests, the installation rate went up to nearly 100% and testing was extremely thorough. The difference was attributed to selecting the test sites carefully (with an emphasis on customer who really wanted the new features), and to following up to make sure that software was actually being installed and used.

Technical support can also provide valuable help to marketing and sales. At SmartCo, support staff were often called upon to identify customers that could serve as sales references. Support people at PlanCo were able to identify follow-up sales opportunities by noticing that large construction management firms often need to transfer plans from place to place. If one location had the software but the other didn’t, that was a sales opportunity. SpreadCo takes calls from customers whose maintenance contracts have lapsed, but then refers them to sales for follow-up. Because technical support is in close contact with the customers, they are in a position to get timely and detailed information that a sales department could not easily get. Likewise, customer contact provides the opportunity to identify market opportunities and trends that transcend specific customers.
7.0 Preliminary Implications for Coordination Theory

Hot line support is a fairly routine process for simple problems. The calls come in, the answers go out. But a hot line takes on a very different quality when a hard problem comes in. Hard problems require the involvement of multiple support people, and they almost always involve multiple exchanges of information between organizations before the problem can be identified. If the problem is defined as a "bug", then the solution involves changes in the software. The change process is predominantly serial, with the problem moving sequentially from customer support to engineering to quality assurance to release management. In the middle lie a range of problems where the individual support person may need to consult others (or do some other kind of research) before providing an answer. One can analyze this as a series of different coordination processes that depend on the complexity of the problem being solved. Figure 2 shows the layering of routines that correspond to different degrees of problem escalation.

Figure 2: Problem escalation hierarchy

<table>
<thead>
<tr>
<th>Problem Difficulty</th>
<th>Number of Call-backs</th>
<th>Number of Supporters</th>
<th>Coordination Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trivial</td>
<td>None</td>
<td>One</td>
<td>Pooled</td>
</tr>
<tr>
<td>Easy</td>
<td>Few</td>
<td>One</td>
<td>Pooled</td>
</tr>
<tr>
<td>Moderate</td>
<td>Few</td>
<td>Few</td>
<td>Distributed search</td>
</tr>
<tr>
<td>Hard, no bug</td>
<td>Many</td>
<td>Many</td>
<td>Distributed search</td>
</tr>
<tr>
<td>Hard, bug</td>
<td>Many</td>
<td>Many</td>
<td>Serial</td>
</tr>
</tbody>
</table>

The terms "pooled" and "serial" in figure 2 are meant to evoke Thompson's (1967) classic discussion of interdependence in organizations. A "pooled" process is appropriate where there is little interdependence (except perhaps
through utilization of common resources). In a hot line, this corresponds to the situation where calls can be handled easily by individuals. A "serial" or sequential process is needed when one step (e.g., recoding) depends on another (e.g., problem diagnosis) and must be completed in a specific order. The term "distributed search" is not from Thompson, and it is meant to imply a different kind of coordination mechanism whereby knowledge is shared among organizational members. A "distributed search" process is necessary when interdependence arises because knowledge is distributed among individuals in the organization.

The sharing or distribution of knowledge within the organization is an important aspect of coordination in a software hot line. In the examples I have described, two general kinds of knowledge distribution can be identified. First, there is "pre-distribution", which is the pre-condition for "pooled" coordination. This phenomenon could be modeled as a diffusion process. Many organizations promote the dissemination of knowledge with training, memos, meetings, face-to-face work settings, and other practices. Depending on the particular kind of knowledge in question, one would expect these efforts to be more or less effective (c.f., Winter, 1986).

Second, there is something that might be called "meta-distribution" of knowledge. The actual knowledge or skill required to accomplish a task is not shared or diffused, but a description of that knowledge is shared so that others know where the skills are located. In computational terms, a "pointer" is established to the knowledge. By "knowing who knows," members of the support group can locate needed expertise very quickly. This form of meta-knowledge is a pre-condition for efficient task assignment in systems with
specialists, as well. In some cases, the surface features of problem descriptions may signify organizational sub-units to which a problem can be referred; in other cases, a deeper understanding of the problem is required. In either case, the actual ability to solve the problem is not required, because the problem can simply be referred to the specialist.

The use of meta-knowledge to locate resources within the organization is a search process. When support people don't know what to do, they ask someone who does. When they don't know who knows, they "ask around" to find someone who does. Search is directed towards somebody who may know or can suggest someone else who will. Analytically, one can distinguish between search (multiple individuals) and no search (single individual). Shared knowledge reduces the need for search, while meta-knowledge facilitates search. Either way, the knowledge gained by individuals within the organization is made collectively available for the benefit of all.

Over time, this collective knowledge and the routines for making it available change. In this sense, the organization learns. One can identify three kinds of learning which occur in the context of technical support: substantive, symbolic, and structural. Substantive learning occurs when a new problem is identified and solved. By adding knowledge to parts of the organization, substantive learning changes the constraints on task assignment. Symbolic learning occurs when the significance of a call, a caller, or a problem area changes (e.g., when an important new product starts having "one too many" problems). Symbolic learning changes the significance of the work for the people involved. Structural learning changes the routines used to conduct work, and therefore the coordination processes themselves.
These categories have been identified in the literature on organizational learning, but usually in isolation. Researchers with an interest in strategic issues tend to emphasize substantive learning (e.g., Fiol and Lyles, 1986; March and Olsen, 1976) cultural theorists emphasize symbolic learning (e.g., Schein, 1985), and organization theorists emphasize structural learning (e.g., Leavitt and March, 1988). In the practical context of hot line support, all three aspect of learning are important and interdependent.

Learning represents an important dimension to coordination theory that we are just beginning to explore. Our analysis to date (e.g., Malone, 1987; Crowston, Malone, and Lin, 1988) has compared static organizational forms at one moment in time. While this may be appropriate for computational models, the ability to learn and change is a crucial aspect of social organizations. Coordination processes in real organizations are not static, so our theory should not be static, either. Recent organizational simulations have yielded some provocative results in this area (Narayanan, 1990). These simulations tested the relative advantage of specialists versus generalists under different assumptions about task assignment, learning, and task load. The results indicate that learning is a vital element in choosing the most efficient organizational form. This finding needs to be pushed further. I am currently planning additional research to explore learning in the technical support context. This research will emphasize the relationship between learning and coordination and will hopefully enrich our understanding of coordination processes in real organizations.
References


<table>
<thead>
<tr>
<th>Acronym</th>
<th>Product Area</th>
<th>Number(^2) Sites/Copies</th>
<th>Number(^3) Products</th>
<th>Number(^4) Platforms</th>
<th>Number Supporters</th>
<th>Organization Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>tCo</td>
<td>Expert system development tools</td>
<td>400/3000</td>
<td>5</td>
<td>10(^{+})</td>
<td>12</td>
<td>Specialists</td>
</tr>
<tr>
<td>Co</td>
<td>Cash management systems for banks</td>
<td>18/18</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>Personal Assignment</td>
</tr>
<tr>
<td>Co</td>
<td>ADA compilers, tools</td>
<td>900/4000</td>
<td>70</td>
<td>10(^{+})</td>
<td>3</td>
<td>Generalists</td>
</tr>
<tr>
<td>Co</td>
<td>Sales/marketing support</td>
<td>5/5</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>Teams of specialists</td>
</tr>
<tr>
<td></td>
<td>Low-end PC database</td>
<td>100,000/300,000</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>Generalists</td>
</tr>
<tr>
<td>Co</td>
<td>Office automation</td>
<td>200/1000</td>
<td>1</td>
<td>12(^{+})</td>
<td>2</td>
<td>Generalists</td>
</tr>
<tr>
<td>Co</td>
<td>Prolog compilers</td>
<td>???/12,000</td>
<td>1</td>
<td>??</td>
<td>1</td>
<td>One person</td>
</tr>
<tr>
<td>Co</td>
<td>Enterprise-wide MIS for manufacturing firms</td>
<td>3500/3500</td>
<td>1</td>
<td>3</td>
<td>43</td>
<td>Specialists</td>
</tr>
<tr>
<td>Co</td>
<td>High performance HW, languages and O/S.</td>
<td>300/300</td>
<td>6</td>
<td>2</td>
<td>8</td>
<td>Specialists</td>
</tr>
<tr>
<td>Co</td>
<td>Decision support tools</td>
<td>150/???</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>Specialists</td>
</tr>
<tr>
<td>dCo</td>
<td>Distributed spreadsheet</td>
<td>40,000/???</td>
<td>1(^{+})</td>
<td>12(^{+})</td>
<td>19</td>
<td>Specialists</td>
</tr>
<tr>
<td>Co</td>
<td>CASE tools</td>
<td>100/300</td>
<td>1(^{+})</td>
<td>1</td>
<td>9</td>
<td>Generalists</td>
</tr>
<tr>
<td>dCo</td>
<td>Project mgmt tools</td>
<td>???/4500</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>Generalists</td>
</tr>
</tbody>
</table>

\(^2\) Approximate number of sites/licensed copies as reported by the interviewee at each site. In cases where question marks appear, the respondent was unable to even guess at a number.

\(^3\) Approximate number of products currently sold. Because some firms offer the "same" product on several different platforms, this number can be misleading.

\(^4\) This is the approximate number of different hardware platform - operating system combinations that each company sells products for. PCs and compatibles running MS-DOS are counted as a single platform. Not all products are available on all platforms.