December 2006

Failure Trap: Cyclical Failures in IS Implementation

Rueylin Hsiao
National University of Singapore and Logistics Institute Asia Pacific

FAILURE TRAP: CYCLICAL FAILURES IN IS IMPLEMENTATION

Social, Behavioral and Organizational Aspect of Information Systems

Hsiao Ruey-Lin
National University of Singapore &
The Logistics Institute of Asia Pacific
(NUS-Georgia Institute of Technology)
bizhr@nus.edu.sg

Abstract

The causes of IS (Information Systems) failure have become increasingly complex. This requires us to be sensitive to how a failure may be situated in the socio-historical context of organizations. This study analyzes how the collapse of organizational learning may be attributed to recurrent IS failures in one particular organization. It suggests that the repeated failed implementations are due to four learning barriers, in terms of the limits of organizational intelligence, disincentive for learning, inappropriate organizational design, and education barriers. These learning dysfunctions may cause organizations to become trapped in permanently failing IS projects. As a result, organizations become entrapped in serial failures when one IS project after another is tried out and then abandoned before enough experience has been accumulated for it to be employed successfully. Therefore, initial failure leads to further search and change, which leads to more failures. This study elaborates the cyclical pattern of IS failure by analyzing the concept of "failure trap", and suggests ways to break the cycles of IS failure.

Keywords: Recurrent IS failures, learning failures, IS implementation, organizational learning

Introduction

The failure of enterprise system remains common despite various causes have been investigated in prior studies, such as organizational politics (Mitev 1996), organizational culture (Romm et al. 1991), institutional theory (Kling and Iacono 1989), conflicting frames (Schön and Rein 1994) and organizational learning (Lyytinen and Robey 1999). The choice of a lens significantly depends on how the way in which IS failure is occurred and observed. Hence, the analysis of IS failure is at times straightforward, for instance if the key problem is a lack of funding, but at other times controversial. The failure of "good" systems may be an early warning of the impact of longstanding problems in organizations and an indication of the powerlessness of certain stakeholders to achieve their goals. Alternatively, the failure of "bad" systems may actually be a good thing for the organization, whereas the success of "good" systems may invariably cause unintended bad consequences for the organization. The subject of IS failure thus requires continual research efforts and begs for analyses that go beyond normal IS failures, which focus more on the technical difficulties within IS implementation (Dalcher, 2003; Sauer, 1999).

Adding to the complexity of these studies, this research reports an alternative pattern of IS failure, based on organizational learning perspective. The failure of IS project, as illustrated in a field study, takes on a life of its own over a long period of time. By analyzing the collapse of organizational learning, this study examines how a firm may be trapped in persisting technology implementation failures.

In the following, the background literature on IS failure is introduced and a theoretical framework based the organizational learning perspective is explained. The methodological issues are then discussed. The research finding

section describes how, in the case of the selected food manufacturing company, problems emerged and evolved through five major time periods. The findings examine how learning dysfunction may contribute to "failure trap" observed in this study. Lastly, theoretical implications and practical considerations are suggested for understanding the cyclical pattern of IS failure.

Theoretical Background

IS failure is a subject receive continuous attention. Lyytinen and Hirschheim (1987) characterized three types of IS failure: process, interaction and correspondence failure. In *process failure*, IS projects are aborted because of undesirable conditions such as schedule slippages, overblown budgets, or financial crises during the implementation stage. *Interaction failure* occurs during or after project completion, and is due to users' non-acceptance because of, for example, lack of motivation or a low level of user involvement. Lastly, in *correspondence failure*, IS projects are completed but fail to meet specific objectives such as monetary saving, improved efficiency and greater productivity

Previous studies have predominately examined IS failure by identifying inhibiting factors. For instance, DeLone and McLean (1992) summarized current literatures and identified six main categories of problems that cause IS failure: system quality, information quality, IS usage, user satisfaction, individual impact, and organizational impact. In a similar way, Sauer (1999) reviewed more than thirty related studies and summarized them into 12 categories of factors to which IS failure may be attributed. More recent literatures extended the technical emphasis to consider the social implication on technology implementation. Robey and Boudreau (1999), for example, suggested that organizational politics, organizational culture, institutional theory, and organizational learning. This study is anchored in the lens of organizational learning.

In the perspective organizational learning, to function effectively, organizations need to learn from previous experience, to exploit experiences from other organizations, and to develop ways to interpret those experiences (Levitt and March 1988; March and Olsen 1975). IS projects may fail because an organization fails to learn. The current studies identifies four possible sources of learning barriers (based on the review in Lyytinen and Robey 1999): limits on organizational intelligence, disincentive for learning, organizational design, and education barriers.

Limits on organizational intelligence. The ability to learn is limited by four reasons. First, learning dysfunction may caused by bounds on the capacity to process information and to make sense of experience (Simon 1990). For example, information overload can confuse decision makers and lead to the choice of an incorrect course of action. Second, high staff turnover can cause the loss of cumulative organizational memory so that organizations are unable to transfer experience from one project to another (Carley 1992). A stressful working environment may leave organizational members no time to examine the basic assumptions underlying system implementation. Third, employees are constrained by pre-exiting institutional arrangements and thus unable to learn from each other. Fourth, organizational members often draw invalid conclusion due to their insensitivity to local context. An earlier study of federal program failure in the USA offers a useful reference point. Derthick (1972) investigated seven federal urban housing programs in the 1970s, none of which achieved more than minimal success, and all of which failed as measured against the objective of constructing the expected quantities of housing. Derthick concluded that these failures resulted mainly from the limited ability of the Federal Government to influence the actions of local governments. First, the Federal Government had limited knowledge of local contexts. Federal officials were unable to incorporate necessary adjustments to local interests. Secondly, federal officials had limited use of local support to outweigh opposition, which resulted in poor coordination of complex projects. Thirdly, federal officials believed that they had a duty to develop a "demonstration" that could be used as a model to show other local governments what ought to be done. Hence, they ended up setting up plans that were too ambitious. In striving for idealized goals, they achieved nothing at all.

Disincentive for learning. An obsession with past success and fear of punishment may discourage learning so that organizational members are motivated to wipe failed experiences from the organizational memory. In this situation, learning occurs but is forgotten or not kept for later use (Argyris and Schön, 1996: 85-107) Such defensive routines manifested themselves in secret maneuvering (against the implementer), empire building to guard against threats, political coalition, and cover-up actions to avoid embarrassment and the admission of failure (Henfridsson and Söderholm 2000). Wastell (1999) examined three types of social defense (organizational ritual, sibling horde, paranoid isolationism), and attributed IS failure to "the inherently high levels of stress and anxiety that imbue information systems development and that elicit defense-avoidance behavior patterns in project teams." The operation of these social defenses can paralyze the learning processes that are critical to effective IS development.

Organizational design. Departmental boundaries can induce a "walled-off" effect. For example, a traditionally-structured IS department may compete for resources and foster organizational politics and distrust. As a result, the department may become isolated from the rest of the business. The compartmentalized organizational structure may also form islands of organizational knowledge, resulting in poor collaboration and little information sharing (Robey and Newman 1996). The implementer may also be blinded by pre-existing institutional arrangements, which condition the organizational member's specialization and actions (Kling and Iacono 1989). The success or failure of system implementations depends significantly on the coordination mechanism embedded in the institutional structure.

Education barriers. Although it is widely understood that the implementer needs to offer sufficient user training, in practice very few educational schemes are arranged. As Attewell (1992) noted, although firms rapidly bring in powerful information systems, the knowledge needed to use such systems is acquired much more slowly and with considerably more difficulties. The Users may have to spend several years developing an understanding of the strengths and weakness of the technology and its associated new knowledge before assimilating the systems into their work routines. Robey, Ross and Boudreau (2002) highlighted two education barriers to enterprise systems use. The first barrier is related to "configuration knowledge" (learning about the system features), and the second is associated with "assimilation knowledge" (learning new work processes induced by technology). Without configuration knowledge, the users cannot exploit the systems to its full potential; without assimilation knowledge, the users cannot cope with the new organizational structure, managerial process, and responsibilities assumed by information systems.

In this view, IS failures may persist because of a vicious circle of learning failure (Akkermans and van Helden 2002; Lyytinen and Robey 1999). First, the emergence of learning barriers force organizational members to adhere to invalid frames-of-reference, thereby inhibiting learning functions. Second, learning barriers may prohibit members from learning collectively within the organization. Third, learning dysfunctions lead to IS failures, which further enforce learning barriers. As a result, IS failures may ensue because organizations are unable to establish an effective mechanism of detecting and correcting errors (Argyris and Schön 1996).

Although the current studies have highlighted the influence of learning dysfunctions on IS failures, the existing discussion has not yet had a chance to examine how learning barriers may play out in an organizational change process. March and Olsen (1975) summed up that learning barriers can lead to the "failure trap" effect in which organizations can get caught in serial failures when one idea after another is adopted and abandoned before enough experience has been accumulated for it to be used successfully. As a result, one failure leads to further search and change, leading to more failures. Nevertheless, this insight has not yet been fully explored in the extant theories. In IS literature, an important aspect of organizational learning analysis is to investigate how learning dysfunction may evolve and reoccur in the implementation process (cf. Montealegre and Keil 2000; Senge 1990; Sterman, Repenning and Kofman 1997). For this reason, the present study seeks to elaborate the dynamics of failure trap within information systems implementation.

Research Methods

This study employs the organizational learning perspective as the interpretive lens to analyze the cyclical pattern of IS failure (since what people learn, remember, and forget is highly subjective). For an interpretive study, it is not a matter of choosing the correct lens but of following theoretical routes that assist in shedding light on interesting aspects of the question under consideration (Walsham 1995). The purpose is to offer a reciprocal elaboration of the empirical evidence (IS failures) and theoretical conception (organizational learning), so that we can make sense of the social phenomenon as well as understand better the theoretical ideas in their original field (Klein and Myers 1999).

Research Site Selection

The selected firm, FoodCo (a disguised name), was accessed through a referral contact in Taiwan in 1997. The firm was then preparing a major transition from in-house IS development to the implementation of the Oracle ERP (Enterprise Resources Planning) system. Although the initial concern was to analyze the major technology-induced organizational change, when tracing back the history of IS implementation, surprisingly I found that the company

had previously abandoned four major IS projects. I undertook a chronological analysis of FoodCo's technology and organizational problems evolving over different projects.

This case was selected because it potentially provides a rich context to understand learning dysfunction and IS failures, for five reasons: (1) the project ran over a long period of time during which project abortions were clearly distinguishable; (2) similar IS implementation problems (e.g. lack of user commitment) and organizational issues (e.g. lack of process improvement and major political struggles) were recurrent in the organization; (3) IS abandonment and waste of investment had been going on for years without being noticed by the top management; (4) there were multiple interpretations of success/failure concerning the various IS projects (e.g. the implementer perceived the IS projects to be a cumulative success, while the users considered them to be serial failures); (5) the turnaround of IS implementation by early 2004 (before I left the field) was still undetermined. This case aptly illustrates a cyclical pattern of IS failure characterized by a complex interaction of social action and learning dysfunction.

Data Collection

In this study, the main data gathering technique consisted of personal interviews with social actors involved in the IS implementation in different periods of time. To highlight the social dynamics of the recurrent IS failures, I collected data from the implementers (STRAT/ TECH Divisions) and the users (mainly BIZ and DISTRI), as shown in Figure 1. Among the interviews with staff in each department, I selected nine key actors for detailed analysis (following Robey and Newman 1996), because they made critical decisions and directed the course of action in the firm. Their disguised names are: Kevin and David (head of BIZ, in Period 1-4 and Period 5), Irene (manager of BIZ in Period 2-4), Sherry (manager of DISTRI in Period 5), Henry (head of STRAT in Period 1), James (head of STRAT in Periods 2-4), Hiro (head of STRAT in Period 5), Stephen (the mastermind of STRAT in Periods 2-5) and Raymond (head of TECH in Periods 2-5).

[Insert Figure 1 about here]

I flew to Taiwan and conducted all personal interviews with these informants over a period of six years (1999-2004), as shown in Table 1. I interviewed key actors and others in these divisions to trace the developmental process retrospectively (Glick et al. 1990). Although critical activities in the implementation processes were captured as completely as possible, the shortcomings of this technique included the social actor's inability to recall the exact events. To minimize this problem, participatory observations of project implementation within BIZ and DISTRI were arranged in four time periods: April 1998, May 1999, July 2000, June 2001 (each visit spanning one week). For example, I joined many internal activities such as playing poker (a ritual event in BIZ during lunch breaks) and going out with product managers to visit distribution channels. On one visit to a distribution center in Taipei, I was also invited to socialize with the staff and walk through the distribution process. This enabled me to understand their experience with the system.

During the site visits, I attended departmental meetings and participated in many informal conversations in order to appreciate the organizational climate. These interviews and other project archives were useful in constructing the main storyline of this case. I also maintained a close relationship with several social actors in order to keep track of progress. They helped me to acclimatize to the organizational culture, avoid political currents, and become connected with various key actors. However, it became increasingly difficult to gain access to subjects as time went by. As many staff left the company (or were transferred to another department), the "new" people there basically had no memory of the previous IS implementations. I waited until 2003 (February) to regain access to the sites, before deciding that data collection had reached a saturation point and no further fieldwork would be necessary.

[Insert Table 1 here]

During a typical interview session, the informants were invited to answer informally the following questions (in Mandarin), with a focus on the four learning barriers: What have been the roles and responsibilities in your personal career in this company? What were the major activities involved in the recent use of information systems? What problems did you encounter? In your view, why did these problems arise? I translated the transcripts into English with the assistance of a professional language editor (British) to ensure the meaning was conveyed effectively.

Data Analysis

I consulted the contextualist method to analyze data (Pettigrew 1990), which examines the history of emergent changes in an organization, shaped by the social and organizational context. I decomposed data into successive periods in its natural occurrence. My main interest was to examine how learning processes account for the empirical pattern of recurrent failures. As shown in Table 2, I analyzed the case with regard to four main constructs: context (of organization), content (of technology and organizational change), process (of implementation), and outcome (of IS failure).

The data analysis was conducted in a reiterative and emergent process, in three main stages. Initially, the objective was mainly to describe the IS implementation experience in the selected case company, as its problems exhibited useful insights to illustrate technical complexity, organizational politics, and general organizational issues. However, due to logistics problem (I flew in the case site in Taiwan from UK and Singapore in various stages), I had to collect data in different periods. This challenge however let me encounter FoodCo's many attempts to implement the enterprise system, and allowed me to observe a similar pattern of IS failure at each implementation phase. When I analyzed the process data, I began to see the continuity of IS failures. The data analysis was interwoven with theoretical review. I then zoomed in my literature review to the social implication on IS and later to organizational learning. The unique learning process observed in the case firm alerted me to investigate how learning dysfunction may lead to recurrent IS implementation failures (March and Olsen 1975; Montealegre and Keil 2000).

In stage 1, as I analyzed the process data, I tried to identify the boundary of each implementation phase. I demarcated each phase according to three different types of IS failure, in terms of process, interaction and correspondence failure (Lyytinen and Hirschheim 1987). Also, a complete phase of implementation should also include a formal closure of the initiated project. This involved, for example, a transition of consulting firm, technology vendor, and internal stakeholders (e.g. having a new project leader). The outcome is Table 2.

In stage 2, alongside with field reentry, I recoded the process data and examined the learning-related problems by three key questions/constructs (Bacharach et al. 1996): (1) how was the problem initiated? (2) how was the problem perceived (i.e. how did the social actor make sense of the problem) (3) what were actions taken? (4) what organizational learning issues might be observed? (5) what was the outcome of the actions? This analysis helped surface learning issues. For example, a "mismanaged contract" problem at first glimpse may look like a technical failure but underlying the action may be a series of wrong decisions supported by perfect rationale. By tracing how the social actor's make sense of issue, I was able to examine how a problem was misinterpreted by the actor and a misaligned action was taken as a result. This also helped explain why the smart people might make obvious and serious mistakes in a complex IS implementation environment. The outcome of this stage is Table 3.

In stage 3, the story was taken back to the key social actors (who supported this project). I only selected members who had no direct conflict of interest to the IS implementation. Their inputs to the process story were further incorporated into the analysis. Finally, I also invited five experts in Asia Pacific regions (two management consultants, one technical consultant, one senior executive who is responsible to a multi-national corporation IS functions, and one lawyer who supervises large-scale IS contract) to comments on the case. Surprisingly, they all experienced similar learning problems in their previous professional engagement. Their inputs helped enhance the generalizability of the data analysis. The outcome is Table 4.

Research Findings

The findings are divided into six sub-sections. The first describes the company background. The following sub-sections report the social dynamics of IS failures in different time periods in chronological sequence. The focus is on explaining how problems were perceived and activated by different groups of social actors in the changing organizational context over time (see Table 2). A cross-analysis of learning barriers is presented in the next section.

[Insert Table 2 about here]

Company Background

FoodCo is one of the leading manufacturers in the consumer food industry in Asia. Since its establishment in 1967, the company has grown into an international organization employing more than 6,000 people in the core businesses.

It has 52 factories in China and Taiwan and strategic alliances with over 70 internationally known firms (up to the year 2000). Its core business scope ranges from animal foods (e.g. stock feed) to consumer foods (such as plain flour, meat, frozen foods, and beverages), convenience vendor stores, supermarkets, distribution, and other investments. The present study focuses on the consumer foods business in Taiwan (with a revenue of circa US$10.6 billion in 2002).

Until 1989 FoodCo have not yet invested in information systems. The increasing inventory costs and inability to respond to customer demands alerted the top management to pay attention to IS implementation. Moreover, as FoodCo is moving its manufacturing base to mainland China and expanding its sales globally, the use of information system become a top priority in the company. For example, a simple monthly financial consolidation would take as long as 3-4 weeks, which could easily upset audit departments, stockholders and government agencies.

Hence, to support future growth, FoodCo had invested in a series of IS initiatives during the period 1989-2003. However, during this 13-year period, the implementations of these IS projects in BIZ (which is a sales company with more than 300 staff) proved unsuccessful. In spite of continuous investment in various IS initiatives (amounting to more than $5 million in Periods 4 and 5), technical difficulties and organizational problems persisted, and IS projects were aborted one after another.

Period 1 (1989-1990): Initiating IS Outsourcing

In the early stage of computerization, FoodCo established a channel consisting of 14 retailers equipped with POS (Point of Sales) systems. In 1989, Henry (head of STRAT) assumed responsibility for masterminding the IS project and aimed to streamline management processes through computerization. One major decision was to assess whether to keep the IS development in-house or outsource it to an external vendor. Henry concluded that there was insufficient IS expertise within FoodCo and decided to outsource the IS project to a quasi-governmental research institute, IndusCo (a disguised name). Over two years, IndusCo failed to deliver the system on schedule, and the poor software quality also resulted in system breakdowns. The project was aborted by the end of 1990. Henry concluded that it had been the wrong decision to outsource the IS project to vendors who might have technological expertise but would lack an understanding of the company. As he explained:

> *The task of information system development is critical for our company. The project requires a detailed understanding of our management processes. The consultants are familiar with the system development but it may take years for them to fully understand our managerial systems.*

IndusCo's managers did not feel comfortable with Henry's assessment. They responded that the failed project was due to FoodCo's inability to articulate its information needs and changing requirements, which made system analysis and programming extremely difficult. They pointed out that many of FoodCo's processes were inefficient and contradicted the managerial process supported by IS principles (which required the effective horizontal integration of information flow). In addition, IndusCo found that the majority of staff in the sales company (BIZ) had little understanding of, and interest in, computers.

For Henry, however, an important lesson gained from this failure was that outsourcing information systems to external vendors is risky. He perceived that IS development must be kept in-house. Before too long, Henry was urged to take action on further IS initiatives. This time, he proposed to collaborate with Raymond, the head of TECH.

Period 2 (1991-1994): Developing system In-house

Beginning in 1991, FoodCo introduced the new role of "product manager" with the aim of transforming traditional sales representatives to facilitate horizontal product management. Product managers had to gather the client-side responses (from retailers and chain stores) and collaborate with the Research Division to develop new products. Raymond chose WANG systems (hardware) and the COBOL computing language (software) to develop the systems. The scope of this project was extended to cover functional areas such as accounting, ordering management and inventory control.

In 1992, James replaced Henry as the head of STRAT. Stephen (with an American MBA under his belt) was appointed as "mastermind" of STRAT. He regarded the in-house IS development as a core-competence building exercise and had an important agenda: to involve users (BIZ) in the system development. He asked product managers to assist with the software development. But most product managers were worried that TECH seemed to be developing a static system which did not match their dynamic working routine. To cope with the increasingly competitive food and beverage market, FoodCo had to ensure that there were more new products each quarter, and to respond to constant changes of price in different promotional seasons. Irene, who was a senior product manager playing a key role in coordinating the IS development in BIZ, explained:

> *It was a disappointment that the software was not flexible enough to manage the changing product portfolios and pricing fluctuations. The system is in fact a barrier to our work in responding to dynamic market situations.*

Product managers generally perceived that supply chain problems were due to three primary causes. First, FoodCo's ordering system was based on more than 272 administrative forms to manage orders from different channels (using fax machines). In such circumstances, if the outmoded processes were not improved, the systems could only improve the efficiency and not the effectiveness of the order processing. Secondly, the current retailer management (based on traditional channels) also required modifications to complement the modernized channels (such as the convenience vendor stores, or CVS). Thirdly, a new distribution strategy might be needed to respond to a shift in the customer demographic in Taiwan, as more demands were emerging from suburban areas. Therefore, systems designed to automate existing processes might contribute little to the solution of their problems. Kevin expressed a common anxiety towards the IS project:

> *The real problem for us is concerned with retailing conflicts within the channel. We have traditional grandma shops and modern convenience vendor stores coexisting at the same time. Different divisions have different retailing policies, and often these policies are in conflict with each other. The payment terms used for grandma shops are not applicable to the convenience vendor stores. More importantly, if you develop systems without first rationalizing the processes, you end up dealing with three types of complexity: the processes are getting more complex; the systems are becoming more complex; using the systems to help our work processes is even more complex.*

Stephen noted these complaints from Raymond's feedback. He analyzed the situation and concluded that consensus building was needed so that "people will become more cooperative." In 1993, a consulting firm was invited to organize a one-week consensus-building exercise, called a "strategic meeting". Most of the senior executives from various divisions (not product managers) were involved in designing the strategy for FoodCo's future vision. Stephen hoped that the process of consensus building would lead to a coherent strategy and help executives to see the "big picture". In this way, the IS development project might be better aligned with the corporate strategy. Nevertheless, after the strategic meeting, Stephen still did not receive the support of BIZ. Product managers were simply making all sorts of excuses to avoid system analysis tasks and sending junior staff to deal with the analysis of user requirements. Kevin noted:

> *The strategy-making exercise is systematic but is very naïve for us. How can we set a static strategic position and then try to see how information systems can fit in? We are in an industry where competition is dynamic and intensive; most times you just don't have any strategy.*

System development did not go smoothly. As more schedule slippage occurred, the project was overhauled at the end of 1994. However, Raymond continued to let his team work on system specification in isolation from BIZ. He still perceived it as a successful project, as his team had completed the coding for several modules (even though they were not yet online). Stephen, on the other hand, believed that the IS project was simply suffering from teething problems, which could be resolved once further organizational changes were in place. He explained:

FoodCo has a conservative culture. The IS project presents a radical change to most product managers. This is typical. As we recruit more internal change champions to help users understand the system functions, they will begin to see the benefits.

Period 3 (1995-1996): Restarting In-house IS Development

By early 1995, Stephen intended to restart the IS development. However, at the same time, James had a more worrisome agenda in response to the new CEO's emphasis on cultural innovation. The CEO, appointed at the end of 1989, recognized that the old "hard-work" culture was not necessarily helpful in preparing FoodCo for the next decade. His ambition was to maintain that culture and at the same time develop a blended culture of innovation, as he remarked:

We need to keep the traditional spirit of hard-work in jobs, honesty and integrity. But to face the future challenge, we need not only to work hard to maintain our productivity, but we also have to develop a sense of innovation, a culture that can help us to transform from "working hard" to "working smart".

The mission of equipping the firm with a "smart-work" culture became the heart of the next IS development. James had to find alternative solutions to realize a culture consisting of both hard-work and smart-work. As the previous stage of IS development had not progressed smoothly, James asked Stephen and Raymond to search for a technical solution that would help staff demonstrate smart working. The solution was to deploy a "smart machine" by installing groupware (Lotus Notes) to enhance internal communication. They hoped that the system would demonstrate to the CEO various sensible ways of smart working (e.g. better communication via emails and knowledge sharing). Stephen explained the benefits of this "smart machine":

The groupware will promote interdepartmental communications and collaboration. This in turn will encourage user participation as more product managers begin to be convinced by the system… We can also demonstrate to the divisions how information technologies can best be deployed.

The adoption of groupware did not in practice lead to a flourishing of the "smart-work" culture. A negative view by the hard-work culture could be observed in the regular cross-departmental "coordination meetings". These meetings typically involved about 20 personnel from different divisions (BIZ, DISTRI and MANUFACT). At each meeting, the chair would check a follow-up list for a particular problem with one division at a time, while the other staff sat by idly, scribbling proposals or making occasional phone calls outside the room. Sometimes the chair's attention would be fixed on one difficult issue for a long time. At other times, staff began to chat about private matters. The meetings often lasted between five and six hours every week. Employees expressed their frustration at having to attend such meetings, but would feel that they were not working hard enough if they were absent. Irene explained:

We all know these meetings are hopeless. There must be a better way to coordinate! But it is our tradition to continue this type of meeting… We have one on Monday, one on Thursday, and sometimes another one on Saturday. We waste a lot of time in these meetings without getting anything solved, but it is a method of impression management. You have to show your loyalty to your bosses by showing up at these meetings.

Kevin and product managers could not see how the "smart-work" culture could be achieved by the application of the groupware in the current context. Apart from the tedious meetings, product managers were more worried about day-to-day problem solving with regard to different channels, new product launches, and emerging competition. As users' complaints mounted, Stephen realized that the "smart machine" metaphor was beginning to seem rather hollow. However, he and Raymond were still under pressure to deliver systems that could inspire a "smart-work" culture. They decided to re-focus their attention on in-house IS development in order to deliver the "smart machine" vision. In early 1996, Raymond redirected resources to system development and attempted to speed up the implementation, though still working in isolation from the users.

Raymond's team was facing an uphill struggle in developing the system. They still could not persuade BIZ to send senior product managers to participate in system design meetings, as they were engaged in too many coordination meetings. As programmers incorporated more and more suggestions from junior product managers, the system became increasingly unstable. Raymond was unwilling to unveil these systems for fear of receiving another outpouring of grievances from users. He asked Stephen to organize educational programs for these junior product managers in the hope of improving their input on system requirements. Stephen replied:

> *We have no time for such training. The system has been delayed for some time. We must catch up with the schedule. These are just glitches in most implementation projects – the price of progress.*

Stephen suggested speeding up the developmental processes and produced a prototype for users in an attempt to win back their confidence in this project. By late 1996, several order-processing systems had already been completed and implemented in BIZ and DISTRI. To ensure users' commitment to system usage, Stephen organized an auditing team to check whether the systems were used in each site. The audit report indicated that more than 90% of the seven test sites were online, and product managers had begun to use the systems to process orders. Stephen and Raymond regarded this as a great success after all the hard work they had put into system development.

But users had a different perception. They considered the systems incomplete and unworkable for their practical needs. For example, the systems could not produce a correct inventory report on a weekly basis. Senior product managers could not relate the manual bookkeeping to the data entry enabled by the computerized system. The product items offered by the systems were often inconsistent with the actual product categories.

To cope with system usage auditing, BIZ and DISTRI decided to produce a "demo" system secretly. They put mock-up data into the systems in order to show the auditor that the systems were in effect online. Several sites in BIZ and DISTRI decided to launch their own systems. For example, one site used Word and Excel to produce semi-manual inventory reports, one site privately hired a programmer to design a standalone system to handle data entry and generate reports, and another site purchased packaged software to deal with daily order management. Kevin was upset by the rushed implementation and audit practice, to which he had not given his approval. He regarded the project as a major failure: the systems failed to support product managers' routine functions, produced more confusion, and added to their workload. Irene explained:

> *You cannot design the ordering and inventory system without considering the manufacturing and distribution operations. The current systems only deal with data entry and reporting. Even though the systems can generate accurate outputs, they don't, and they fail to consider capacity planning, demand forecasting, and resource allocation as integrated activities. Without such functions, the systems are nothing but a big calculator – and an expensive one.*

Period 4 (1997-1998): Renewing Hardware and Software

In 1996, FoodCo undertook a major restructuring (Figure 1). This reorganization aimed to increase local autonomy in individual divisions. In BIZ, the First Food Division included traditional businesses such as livestock, flour, edible oil and animal feed, which were located in Indonesia and six other major cities in China. The Second Food Division was regrouped with a focus on beverage and vending machines. The Third Food Division encompassed major business lines such as soy sauce condiments, health foods, baking, chain stores, frozen foods, dairy and meat. In DISTRI, one division handled products that could be delivered at a normal temperature (25 degrees centigrade) and another managed the logistics of cold storage and frozen foods.

During mid-1997, TECH redesigned some of the systems and began to install them for users in BIZ. Not too long after, users began to experience unstable systems and frequent program downtime and asked for more system modifications. Some product managers began to grumble openly about the inconvenience caused by the systems. Programmers explained that the situation was not entirely their fault but was in part due to the changing expectations of users. Many product managers simply chose not to use the systems. Irene and her team experienced most of these difficulties, as she commented:

> *When you have a virus [this refers to virus in general sense, i.e. the inappropriate design of information systems, not computer virus] in the system, the more widely the systems are used, the more difficult it is to control the outbreak. These programmers have no idea what is the real life of sales, marketing and channel. They are putting a virus into the systems and the outbreak is the result.*

Soon afterwards (in 1998), product managers began to ignore the IS-related activities again and turned to an emergent problem known as "premature product death" in the company: as many new products were launched at the same time, the distribution systems failed to deliver these products on time in accordance with the marketing promotions. Therefore, customers could not buy these new products in retail stores, and this resulted in abandoning new products that might have had significant market potential. As the "premature product death" problem escalated, most product managers aborted the system development tasks. Although Stephen requested product managers to rejoin system development, no one turned up at any further system design meetings. Kevin noted:

> *Premature product death is a problem of the supply chain. We are weak in coordinating supply activities. Simply dumping systems on us will not cure the problem. The outmoded working processes require a substantial revamp. Otherwise, getting my people to develop the systems will only add to their already exhausting workload.*

At the same time, FoodCo's organizational climate became even bleaker. Premature product death had caused low staff morale. In BIZ, the increasing workloads from daily firefighting and system development discouraged many product managers. By this time, the company's competitors aggressively recruited FoodCo's product managers. The steady loss of senior staff meant the stagnation of accumulated industrial knowledge. There was not enough time for new staff to assimilate the valuable supply chain knowledge from senior product managers.

On the other hand, the "premature product death" problem alerted Raymond to the supply chain difficulties which had not been included in the initial system design. At this time, Kevin was replaced by David as the head of BIZ. James was transferred to oversee operations in China, and Hiro, who had been with BIZ for more than ten years, took over STRAT (at the end of 1998). His plan was to implement a system that could resolve the "premature product death" problem in the supply chain.

Period 5 (1999-2003): Best Practice Transfer via Oracle ERP Systems

By 1999, in the domestic arena, FoodCo was also moving toward a different business orientation: from forecast-based production to demand-based production. This placed more emphasis on supply chain management. Meanwhile, the CEO had lost interest in promoting the "smart-work" culture and turned to other public relations and marketing activities.

Hiro was attracted to Stephen's proposal of transferring "best practices" to resolve FoodCo's operational problems. The idea was that several leading software packages, such as SAP and Oracle's ERP systems, had been built by adopting the best practice of Fortune 500 American companies. If FoodCo could implement one of these packages, it would be able to implement not only the systems but also transfer the best practices embedded in the systems. An extensive survey was conducted by Stephen to evaluate this idea. The short-list included SAP and Oracle. Stephen and Raymond suggested implementing Oracle ERP for two reasons.

First, TECH's staffs were more familiar with the Oracle systems, as they had experience of Oracle RDBMS from previous system development. From a technical perspective, Raymond's team believed that the Oracle systems offered more flexibility for software modification. Secondly, on a trip to Oracle's USA headquarters, Stephen was impressed by how the system was used by other food companies (such as Kellogg). He considered that the Oracle systems would help FoodCo transfer the best practices developed within these USA firms.

The implementation began in early 2000 and the contract was awarded to ConsultSys (an international consulting firm - disguised name). A group of technical consultants was assigned to assist with the initial implementation of the order-processing and inventory management modules. These two modules aimed to automate order entry and integrate the production process among BIZ, MANUFACT and DISTRI. Stephen was afraid that users might again resist using the system. Therefore, he ordered a system "switch-over" instruction: once the Oracle systems had been implemented, all manual operations would be terminated.

The switch-over caused significant operational disruption. Product managers had more problem-solving to do, because MANUFACT did not produce the requested amount of products and many finished products were not allowed to be delivered by DISTRI due to a lack of computer authorization from the Oracle systems. This was because the software demanded that products released from factories needed to match the computer-generated invoices. In reality, MANUFACT had its production plan scheduled in a dynamic fashion, resulting in information discrepancies between the system, the sales company, and the manufacturing sites. Product managers could not get their products out of the factory in time to meet delivery schedules, because the actual production volume was often not consistent with the computer-generated invoice.

The switch-over to the Oracle systems seriously affected the sales company in two major ways. First, FoodCo had to pay the convenient vendor stores a huge penalty if the delivery was delayed. This would also increase the risk of "premature product death" if the delivery was a batch of cold storage products (which were time-sensitive). Product managers had to ask the Vice President of BIZ to sign a "special pass" to get their products delivered, resulting in chaos in distribution operations. Irene noted:

> *The Oracle systems require product managers to prepare numerous order forms for batch data inputs. Our operation was significantly impaired: the distribution of products was delayed because the computer-generated invoices were incorrect and thus the security guards didn't let us get the products out of inventory. Moreover, the computerized system often indicated a zero inventory level, whereas actually the inventory had a pile-up of physical products. We ended up having more "premature product death" cases and penalties because of this delay.*

Secondly, the design principle of the software also constrained the sales and distribution company's daily operations. For instance, the Oracle systems required price setting to be controlled by DISTRI. But FoodCo had long before developed a practice whereby product price was adjusted by BIZ and the compensation of the price difference was charged to DISTRI. The software assumed that changes in product prices were on a regular, predictable basis, but in practice FoodCo had to re-negotiate prices with retailers according to the changing market situation.

The system and its implementation had cost the company more than five million dollars. To maintain budget and demonstrate savings, Stephen decided to reduce the number of operators in each department from an average of six to three persons. Only two operators were sent to Oracle for user training. The plan was for them to transfer their skills to others in each department. But only one of the operators was competent enough to learn all the data entry functions, and she became responsible for all operators' informal training. Stephen believed that users should be able to "pick up" the simple data entry functions.

Since there were very few operators who understood the system functions, the data entry tasks were allocated to only three or four operators in each department. Oracle's system model assumed that the data entry responsibility would be given to product managers, thus spreading the workload. But in FoodCo the enormous volume of order entries ended up being the sole responsibility of the selected operators, who had to work very long hours to enter the data. The data entry tasks were delayed and operators were so tired that they often keyed in incorrect data. Therefore, although MANUFACT had delivered products to the inventory, the computer system often indicated zero stock.

Another data entry problem was that FoodCo had many product categories for each product line. For example, a product manager would typically handle eight different distribution channels and more than 12 product lines (including three or four new products). Each channel had its own price changes according to season. The Oracle systems asked product managers to enter and approve at least 96 forms (8x12) after data entry. Since the product prices were subject to constant market changes, product managers frequently had to re-enter data and re-approve orders. Working with complex product categories, the operators often keyed in erroneous orders and classified them into the wrong categories. Sherry explained the situation with great frustration:

> *The problem is data integrity. You cannot expect the system to function effectively without accurate data. We have not yet standardized our product categories, integrated them with the accounting systems, and performed data cleansing. I dare not imagine what sorts of trouble we will get ourselves into once the data are all on-line.*

Furthermore, the Oracle systems required logging sales data into accounting systems upon delivery. However, this contradicted FoodCo's practice in two ways. First, product managers often dealt with fluctuating sales orders. Often retailers would adjust their orders according to market changes. Since the total quantity of sales orders required constant modification, the Oracle systems seemed to be inflexible. Secondly, the assessment of product defects depended on the retailer side. Retailers might reject a product batch for various reasons. Product managers thus issued invoices until DISTRI settled orders with retailers. But the Oracle systems would not allow this variation and product managers had to revise the computer records (e.g. the quantity of orders) for each retailer. Then, product managers would have to reissue invoices. This not only produced extra workload and confusion for product managers, but there was also a danger that it would arouse the suspicions of the Inland Revenue Service because of the possibility of tax avoidance.

David also requested that ConsultSys modify its system to accommodate their local practices. But the request was not accepted because ConsultSys was unable to alter Oracle's policy to offer a "share service" model – the software should share the same functionalities in each industry in order to standardize best practices embedded in the system. Under pressure, Raymond sent his team on specialized courses to learn how to develop their own Oracle-based systems to meet users' needs, and the TECH staff spent almost a year redesigning interfacing applications to transmit data to the Oracle systems.

Meanwhile, channel managers in DISTRI were frustrated to learn that the Oracle systems covered only about 10 per cent of distribution functions. They expected the Oracle systems to cover functions such as forecasting and replenishment, pricing and revenue optimization, and logistics management. David set up a team led by Sherry to work with LogiSys (disguised name), a firm which specializes in logistics software, to introduce a different system. David's team had begun to work with LogiSys to develop a POC (Proof of Concept) system by the end of 2003. The first test site selected 20 SKUs (Stock Keeping Units) of dairy products and lasted for six weeks.

In January 2004, unaware of the LogiSys project, Stephen announced the next stage of his technology project. He was preparing to introduce the idea of "electronic ordering". The aim was to let retailers place orders directly through the Oracle systems. He estimated that this would promote further efficiency and improved manpower use in the long term. But Sherry was worried that this project would further worsen the data integrity problem, as she stated pessimistically:

> *I'm sure this is going to put us into another cycle of disaster. This is not IS implementation; it's IS kidnapping… I guess the situation we have here is similar to "the emperor's new clothes". I hope that one day there will be someone brave enough to tell Stephen that he isn't wearing anything.*

The Analysis of Failure Trap

The process analysis provides a theory elaboration to the concept of "failure trap" suggested by March and Olsen (1975). Without the process analysis, we may observe individual problems, such as organizational politics, organizational culture, and institutional constraints, in each implementation phase. By investigating the process, we can see learning dysfunctions caused by intertwined organizational problems leading to recurrent IS failures. An initial analysis may show that the recurrent problems were resulted from a series of "bad decisions". But we may wonder why such a series of bad decisions, plus expensive wastage, went on for so many years without being audited. A deeper analysis into organizational learning barriers is useful in understanding how the bad decisions were made and resulted in recurrent organizational problems and persisting IS failures (Table 3).

[Insert Table 3 here]

Limits on Organizational Intelligence

In each phase, the social actors' (the IS project office) reasoning logic was significantly affected by the external and internal environments. This affected their objective assessment of problems at hand and often misinterpreted the situation. Thus actions were taken which mismatched the users' expectation. Another thing inhibiting organizational intelligence was FoodCo's staff turnover. In Periods 1 to 3, the turnover was relatively stable. However, when product managers became demoralized by the "premature product death" problems in the supply chain, they left the

company. The resulting loss of valuable knowledge due to high staff turnover in the supply chain only made system development worse, as more outmoded practices were built into the systems. The outbreak of technical difficulties made the organizational climate ever more stressful and blinded the implementer as well as users.

The stressful situation was made worse by STRAT and TECH's putting forward further technological solutions (RDBMS and Oracle systems). As a result, FoodCo's executives were too busy solving immediate problems in operations to reflect on their ineffective practices. The introduction of various information systems only further limited their capacity to be self-critical (cognitive limitations). Therefore, FoodCo's managers become unable to implement process improvement and frequently made bad decisions. For example, Stephen (STRAT) tried to use groupware to promote the "smart machine" and regain user confidence (Period 2); he cancelled training to avoid project delays (Period 3 and 4); he sought to achieve globalization by using English-version software (Period 5); and he tried to expedite integration by introducing switch-over implementation (Period 5). Raymond (TECH) constantly resorted to more advanced technologies (COBOL, RDBMS and Oracle) to cure technical difficulties. Kevin and David (in BIZ) also made bad decisions as they and their team were buried in outmoded retail policies and unproductive supply chain operations (such as "premature product death").

In the stressful working environment, the loss of organizational memory and information overload (constantly implementing new systems) had constrained organizational intelligence. Under cognitive limitations, FoodCo's managers were making decisions leading to incorrect courses of action and they were unable to transfer their experience (mistakes) from one project to another. They also had no time to examine the basic assumption underlying system implementation. As a result, IS failures persisted.

Disincentive for Learning

The in-fighting and sour relationships between the implementer and users led to the spread of defensive routines. As a result, the distrust further escalated when Stephen failed to engage users in system implementation (Period 3, 4, 5). Moreover, users chose to follow a clandestine system implementation of standalone applications (in Period 4) and LogiSys (in Period 5) rather than negotiating with the implementer in order to improve the ineffective supply chain processes, product strategy, and retailing policies, and this eventually caused more confusion for themselves. Recurrent IS failures ensued. James and Hiro basically handed over their leadership to Stephen. As Stephen's political power grew over time, this discouraged everyone from confronting him directly. This also isolated Stephen from users. While the users' cover-up behavior became more entrenched, Stephen was always kept in the dark, constantly making more bad decisions (cf. "paranoid isolationism" analyzed in Wastell 1999). In these situations, learning from mistakes is easier said than done (a similar observation can be found in a hospital context; Edmondson, 1996).

Stephen's commitment to IS projects escalated in the initial stage as he perceived that the project suffered from a temporary setback due to users' lack of familiarity with the technology (evident in Periods 2 to 4). Ultimately, even though almost everyone else in the company regarded the various IS projects as failures, Raymond tried to justify their existence by implementing more technological initiatives (and claiming them as successes, Periods 2 and 3) and Stephen saw them as the "price of progress and success" (in Periods 3, 5). Attribution egoism, as reported by Brown and Jones (1998), is evident here.

Lastly, the interpersonal conflict was evident in FoodCo throughout the history of system implementation. The top management commitment was absent from the whole implementation process. In particular, the new CEO seemed to place more emphasis on self-image promotion. The heads of STRAT were less involved in the implementation too, which allowed Stephen to take full control of the resources. The implementer (STRAT and TECH) were concerned with technological advancement and impatient with users resistance. The users, out of frustration, simply chose to disengage themselves from those projects (Periods 2 to 5). This prevented the implementer and users from collaborating and learning from previous mistakes.

Organization Design

The centralization of the IS function to STRAT presented another learning barrier to system implementation. FoodCo's coordination mechanism was complex and the business model was dynamic. The centralization of IS resources under STRAT did not support the actual needs of users. In this structure, TECH became a supporting

function. A major problem arose when Stephen decided that STRAT needed to "guide" divisions to achieve organizational change.

If we use Derthick's (1972) insights to reflect on FoodCo's situation, we can see how the centralization of resources became a constraint to IS implementation. Stephen was constantly unable to adjust system modifications to local interests (e.g. by putting dynamic pricing into the system). The ineffective coordination between the implementer and users failed to facilitate the complex integration of technology and organization. Furthermore, Stephen had a tendency to conceive goals in ideal terms as a way to legitimize his actions. He kept inflating the objectives and striving for the ideal. For example, he initiated a series of IS projects under in the names of core competence building (IS in-house development in Period 2), the smart machine (groupware in Period 3), system coherence (replacing legacy systems with RDBMS in Period 4), best practices transfer (Oracle ERP system in Period 5), and electronic ordering (in the future). However, he was insensitive to local knowledge and operations, as demonstrated by the ineffective coordination meetings, "premature product death" problems in the supply chain, staff turnover, and the incompatible practices engendered by the Oracle systems. As a result, as users' distrust grew, it was becoming evermore difficult for Stephen to gain any local support.

Education Barriers

FoodCo's implementer also failed to recognize the importance of user training and to acquire configuration and assimilation knowledge (in the terms of Robey, Ross and Boudreau 2002). Throughout Periods 1 to 5, little effort was given to helping users understand IS applications (configuration knowledge). In Period 1, Henry and his team had insufficient knowledge of system development; the failure of the outsourcing project was thus predictable. Within Periods 2 to 4, since the implementer was engrossed in system development, training needs were never taken into account. In Period 5, the implementer sent only a few operators to learn system functions (note: the implementer only sent programmers for technical training in Period 5). The dissemination of system knowledge relied mainly on the informal training offered by one senior operator.

Especially, in Period 5, FoodCo also failed to reflect on the "assimilation knowledge" embedded in the system. The Oracle systems contained the best practices developed from the USA context, which was significantly different from the Taiwanese context in terms of market scope, channel structure, retailing policy, transportation infrastructure, and the legal system (e.g. taxation and accounting practices). FoodCo failed to examine fully the context-specific knowledge embedded in the system and to make the necessary appropriations. This was further constrained by Oracle's embedded policy on "share service". Ignorance of the knowledge embedded in the system therefore induced constant usage relapses.

The failure to transfer "configuration knowledge" prevented users from using the systems with confidence. Additionally, FoodCo completely ignored the transfer of "assimilation knowledge" to users, which would have required users to learn how the system could best be aligned to the new processes and working behaviors. Again, recurrent IS failures were anticipated.

Research Implications

This research offers two theoretical implications to the IS failure studies in the organizational learning literature in general. The existing literature on IS failure offers extensive documentation of technical and organizational factors contributing to the failure of system implementation (e.g.DeLone and McLean 1992; Sauer 1999). Scholars of the social analysis of computing also analyze IS failure by examining organizational dynamics (e.g. Kling and Iacono 1989; Mitev 1996; Romm et al. 1991). However, little studies have explored how these learning dysfunctions may play out over time in a company (cf. Montealegre and Keil, 2000). Especially, we know relatively little about why these learning dysfunctions reoccur over a long period of time.

This study adds to the existing research efforts and explains how learning processes account for the recurrent pattern of failures over a period of 13 years, characterized by interacting causes, vagaries of social forces, and multiple interpretations of outcome. The cycles of IS failure induced by learning failures provide an empirical elaboration of the "failure trap" effect (March and Olsen 1975): The organization becomes trapped in serial failures when one idea after another is tried out and aborted before enough experience has been accumulated for it to be used successfully. Inevitably, one failure leads to further search and change, which leads to more failures. The cyclical pattern of IS failure reported in this study elaborates this concept.

Furthermore, previous IS-related organizational learning studies have examined (see a review in Robey, Boudreau, and Ross 2000) (1) how IS may be used to support organizational learning (focusing on the antecedence), and (2) how IS may result in effective/ineffective organizational consequences (focusing on the outcome). This study adds to the learning literature by addressing how learning dysfunction may inhibit IS implementation, with a focus on the "process". Although it is beyond the scope of this research to suggest a comprehensive solution for problems facing FoodCo, Table 4 highlights four groups of practical considerations.

[Insert Table 4 here]

In conclusion, the case study illustrates how social actors may be bounded by interrelated actions, the effects of which often take time to play out fully. Although it is possible for them to focus on snapshots of isolated parts of the system, they somehow have difficulties seeing how problems are situated in the broader socio-historical picture. By tracing the chronology of the selected case, this research contributes to the IS failure literature through an analysis of learning barriers (bounded rationality, defensive routines, organizational design, and knowledge barriers) where both problems and their solutions are open for negotiation. The learning dysfunctions help us understand how causes of failure evolve over time, leading to a cyclical pattern of failure. Although such analyses may offer no guarantee of future success, they may provide a means of avoiding some future blunders.

References

Akkermans, H. and van Helden, K. "Vicious and Virtuous Cycles in ERP Implementation: A Case Study of Interrelations between Critical Success Factors," *European Journal of Information Systems* (11), 2002, pp. 35-46.

Argyris, C., and Schön, D. A. *Organizational Learning II: Theory, Method, and Practice*. Addison-Wesley, New York, 1996.

Attewell, P. "Technology Diffusion and Organizational Learning: The Case of Business Computing," *Organization Science* (3:1), 1992, pp. 1-19.

Bacharach, S., Bamberger, B., P. and Sonnenstuhl, W. J. "The Organizational Transformation Process: The Micropolitics of Dissonance Reduction and the Alignment of Logics of Action." *Administrative Science Quarterly* (41), 1996, 477-506.

Brown, A. D., and Jones, M. R. "Doomed to Failure: Narratives of Inevitability and Conspiracy in a Failed IS Project," *Organization Studies* (19:1), 1998, pp. 73-89.

Carley, K. "Organizational Learning and Personnel Turnover," *Organization Science* (3), 1992, pp. 20-46.

Dalcher, D. "Beyond Normal Failures: Dynamic Management of Software Projects," *Technology Analysis and Strategic Management* (15:4) 2003, pp. 421-439.

DeLone, W. H., and McLean, E. R. "Information Systems Success: The Quest for the Dependent Variable," *Information Systems Research* (3:1), 1992, pp. 60-95.

Derthick, M. *New Towns in-Town: Why a Federal Program Failed*. The Urban Institute, Washington, 1972

Edmondson, A. C. "Learning From Mistakes Is Easier Said Than Done: Group and Organizational Influences on the Detection and Correction of Human Error." *Journal of Applied Behavioral Science* (32: 1), 1996, 5-28.

Ewusi-Mensah, K., and Przasnyski, Z. H. "On Information Systems Project Abandonment: An Exploratory Study of Organizational Practice," *MIS Quarterly*, (15: 1), 1991, pp. 67-85.

Gill, T. G. "High-Tech Hidebound: Case Studies of Information Technologies that Inhibited Organizational Learning," *Accounting Management & Information Technologies* (5:2), 1995, pp. 41-60.

Ginzberg, M. J. "Early Diagnosis of MIS Implementation Failure: Promising Results and Unanswered Questions," *Management Science* (27:4), 1981, pp. 459-478.

Glick, W. H., Huber, G. P., Miller, C. C., Doty, D. H., and Sutcliffe, K. M. "Studying Changes in Organizational Design and Effectiveness: Retrospective Event Histories and Periodic Assessment," *Organization Science* (1), 1990, pp. 293-312.

Henfridsson, O., and Söderholm, A. "Barriers to Learning: On Organizational Defenses and Vicious Circles in Technological Adaptation," *Accounting Management & Information Technologies* (10), 2000, pp. 33-51.

Keil, M. "Pulling the Plug: Software Project Management and the Problem of Project Escalation," *MIS Quarterly*, (19:4), 1995, pp. 421-450.

Klein, H. K., and Myers, M. D. "A Set of Principles for Conducting and Evaluating Interpretative Field Studies in Information Systems," *MIS Quarterly* (23:1), 1999, pp. 67-94.

Kling, R., and Iacono, S. "The Institutional Character of Computerized Information Systems," *Information Technology & People* (5:1), 1989, pp. 7-29.

Levitt, B., and March, J. G. "Organizational Learning," *Annual Review of Sociology* (14), 1988, pp. 319-340.

Lyytinen, K. J., and Hirschheim, R. Information Systems Failures: A Survey and Classification of the Empirical Literature. Oxford University Press, Oxford, 1987.

Lyytinen, K., and Robey, D. "Learning Failure in Information Systems Development," *Information Systems Journal* (9), 1999, pp. 85-101.

March, J. G., and Olsen, J. P. "The Uncertainty of the Past: Organizational Learning under Ambiguity," *European Journal of Political Research* (3), 1975, pp. 147-171.

Mitev, N. N. "More than a Failure? The Computerized Reservation Systems at French Railways," *Information Technology & People* (9:4), 1996, pp. 8-19.

Montealegre, R., and Keil, M. "De-escalating Information Technology Projects: Lessons from the Denver International Airport," *MIS Quarterly* (24:3), 2000, pp. 417-447.

Myers, M. D. "A Disaster for Everyone to See: An Interpretative Analysis of a Failed IS Project," *Accounting Management & Information Technology* (4:4), 1995, pp. 185-201.

Pettigrew, A. M. "Longitudinal Field Research on Change: Theory and Practice," *Organization Science* (1:3), 1990, 267-292.

Pentland, B. T. "Information Systems and Organizational Learning: The Social Epistemology of Organizational Knowledge Systems," *Accounting Management & Information Technologies* (5:1), 1995, pp. 1-21.

Robey, D., and Boudreau, M. "Accounting for the Contradictory Organizational Consequences of Information Technology: Theoretical Directions and Methodological Implications," *Information Systems Research* (10:2), 1999, pp. 167-185.

Robey, D., Boudreau, M., and Ross, G. M. "Information Technology and Organizational Learning: A Review and Assessment of Research," *Accounting Management and Information Technologies* (10:2), 2000, pp. 125-155.

Robey, D., and Newman, M. "Sequential Patterns in Information Systems Development: An Application of a Social Process Model," *ACM Transactions on Information Systems* (14:1), 1996, pp. 30-63.

Robey, D., Ross, G. M., and Boudreau, M. "Learning to Implement Enterprise Systems: An Exploratory Study of the Dialectics of Change," *Journal of Management Information Systems* (19:1), 2002, pp. 17-46.

Romm, T., Pliskin, N., Weber, Y., and Lee, A. S. "Identifying Organizational Culture Clash in MIS Implementation: When is It Worth the Effort?" *Information & Management* (21:2), 1991, pp. 99-110.

Sauer, C. "Deciding the Future for IS Failures: Not the Choice You Might Think," In *Rethinking Management* Information *Systems: An Interdisciplinary Perspective*, W. Currie and B. Galliers (eds.), Oxford University Press, Oxford, 1999, pp. 279-309.

Schein, E. H. "Three Cultures of Management: The Key to Organizational Learning," *Sloan Management Review* (38:1), 1996, pp. 9-20.

Schön, D. A., and Rein, M. Frame Reflection: Toward the Resolution of Intractable Policy Controversies, Basic Books, New York, 1994.

Senge, P. The Fifth Discipline: The Art and Practice of the Learning Organization, Doubleday, New York, 1990.

Sterman, J. D., Repenning, N. P. and Kofman, F. "Unanticipated Side Effects of Successful Quality Programs: Exploring a Paradox of Organizational Improvement." *Management Science* 43(4): 1997, pp. 503-521.

Simon, H. A. "Bounded Rationality and Organizational Learning," *Organization Science* (2:1), 1990, pp. 125-134.

Walsham, G. "Interpretive Case Studies in IS Research: Nature and Method," *European Journal of Information Systems* (4), 1995, pp. 74-81.

Wastell, D. G. "Learning Dysfunctions in Information Systems Development: Overcoming the Social Defenses with Transitional Objects," *MIS Quarterly* (23:4), 1999, pp. 581-601.

Table 1. Field Interviews Scheme

Dates	Implementer	Users	Consultants	Total
August 1997 and April 1998	STRAT: 4 TECH: 5	BIZ: 4 DISTRI: 12	9	34
May 1999	STRAT: 3 (one with 3 times) TECH: 1 (4 times)	BIZ: 12 DISTRI: 8 MUNUFAC: 4	4	32
July 2000	STRAT: 3 TECH: 3 (one with 3 times)	BIZ: 5 DISTRI: 4	2	17
June 2001	STRAT: 2 TECH: 1	BIZ: 6 DISTRI: 4	2	15
Nov. 2002	N/A	BIZ: 6 DISTRI: 3	N/A	9
Feb. 2003	STRAT: 2	BIZ: 4 DISTRI: 3	1	10
Dec. 2004	STRAT: 2	BIZ: 7 DISTRI: 2	3	14
Total	26	84	21	131

Note: On average, each interview lasted 1.5 hours.

Table 2. A Process Analysis of Cyclical IS Failures in FoodCo

	Period 1 (1989-1990)	Period 2 (1991-1994)	Period 3 (1995-1996)	Period 4 (1997-1998)	Period 5 (1999-2003)
	IS development via outsourcing	*In-house IS development*	*Continued in-house IS development*	*IS-redevelopment using RDBMS*	*Best practice transfer via the Oracle ERP systems*
Context	The firm was established with a 'hard-working and honesty' culture.	The firm sought regional expansion and introduced the 'product manager' system in order to facilitate horizontal product management.	The problem of operational inefficiency continued. The CEO sought to promote a 'smart-work' culture.	The firm undertook major restructuring and emphasized supply chain operation. The operational difficulties in the supply chain were prominent.	FoodCo was moving toward forecast-based production from a demand-based model.
Content	The firm brought in IndusCo to develop software to automate its retail operations.	Raymond (TECH) replaced the legacy system with the WANG mainframe and employed the COBOL language to develop the applications covering accounting, ordering management, and inventory control. Stephen (STRAT) considered the IS project as a *core-competence building* strategy.	The IS project was reinitiated and the groupware (Lotus Notes) was implemented in tandem. Stephen (STRAT) thought that the *smart machine* concept (i.e. groupware implementation) could help to introduce a 'smart-work' culture and win back users.	Raymond (TECH) upgraded the technological infrastructure using Oracle RDBMS (Relational Data Base Management Systems) in order to achieve system consistency. Stephen (STRAT) agreed that a *technological upgrade* would help to solve the operational problems.	Stephen (STRAT) scrambled to implement the Oracle ERP systems, with the help of ConsultSys, focusing on order and inventory management. The system was implemented to BIZ and DISTRI (excluding MANUFACT). Stephen thought that the supply chain problem could be resolved by *transferring best practices* drawn from Fortune 500 firms.
Process	The firm encountered a series of outsourcing issues, including mismatch of processes, unclear specifications, and a protracted development schedule. The vendor (IndusCo) perceived that FoodCo was unable to articulate its needs and kept changing its system specifications. This made the tasks of system analysis and programming extremely difficult. But the implementer	The users felt that the systems hindered a response to dynamic market situations. Stephen thought that the collaboration problem was due to a lack of coherent strategy and inter-departmental consensus. He organized strategic meetings in the hope of building consensus and aligning strategy to IS development. Although user involvement was considered the most important task, no	Stephen considered that groupware could be used to demonstrate the smart-work culture and educate users in the use of information systems. Under time pressure, Raymond had to expedite the IS development while working in isolation from users. The systems constantly presented technical problems. The users were more worried about day-to-day firefighting in different retail channels, new	TECH faced a vast array of software and hardware problems. The systems were designed without user inputs, resulting in outbreaks of technical difficulties. Product managers were overwhelmed by supply chain problems – known as 'premature product death'. They gave up supporting IS development tasks and concentrated on stabilizing their supply chain	The radical changeover to the Oracle ERP system caused a series of problems, including operational breakdowns, data entry delay, mismatched inventory records, and work overload for users. Users were unable to operate under the new system due to lack of training and language difficulties. STRAT and TECH responded by sending technical staffs for intensive training. TECH

	(Henry) considered it to be a risky decision to outsource core business to external vendors.	training was organized. The political conflicts and distrustful relationship between STRAT and BIZ grew.	product launches, and emerging competitions. Stephen ignored user training in the hope of making up for schedule delays.	operations. Stephen avoided user training in order to expedite the IS development. The relationship between STRAT and BIZ become sour.	began to design supplementary systems for the users. The users regarded the system as impractical. The distrustful relationship deepened.
Outcome	*Process and termination failure:* The systems did not reach users. The outsourced IS project was eventually abandoned.	*Interaction failure:* Users resisted participating in IS development. The systems were stalled. But Raymond considered it a success, as most system design tasks were completed on schedule (but not yet on-line).	*Interaction and termination failure*: Users did not accept the groupware. IS development was suspended owing to the users' inability to support system analysis tasks.	*Correspondence and termination failure*: Users rejected the systems and turned to PC-based package software and began to develop their own systems individually. The in-house IS project was terminated.	*Interaction failure*: Users decided to abandon the Oracle systems and launch a different system (LogiSys) on their own initiative. But Stephen declared the Oracle implementation a success.

Table 3. The Dynamics of Failure Trap

	Period 1 (1989-1990)	Period 2 (1991-1994)	Period 3 (1995-1996)	Period 4 (1997-1998)	Period 5 (1999-2003)
Major theme	Initiating IS outsourcing	Developing supply chain operation system	Restarting IS development	Escalating supply chain problems	Transferring the embedded "best practices"
Problem encountered (as observed by the researcher and technical expert)	In experienced staff were assigned to handle IS projects, who were unable to articulate system specifications. Business processes were not improved before IS implementation. User training was neglected because the project office's energy was absorbed by the dispute with the outsourcing vendor.	Retail policies were outmoded. IS staff were interested only in technical development and had little experience in the retail channel. Business processes were still not improved. The new product manager system made the processes more sophisticated. User training was ignored due to the emphasis on "strategic meeting".	The junior product managers were giving wrong specification to the system developers. No user communication was provided to verify system requirement. The system did not match the user's supply chain requirement. User training was sacrificed because of time pressure and distraction from "smart-work" culture initiative.	As the system was widely implemented, more users' tasks were disrupted by the misaligned system and asked for more modification. The outmoded processes were still not improved, leading to more channel management problems, as corporate restructuring occupied most people's attention. .	The package system required significant localization for effective use.
Problem perceived by actors	The technical consultants lacked in-depth understanding of the company.	The conflict between the IS project office and retail channel was escalating, inhibiting an in-depth understanding of the company's operation.	The internal conflict could be resolved by building a new organizational culture (smart work). Users' inertia was too high, which requires radical actions.	The technical experts assessed that outdated hardware and software impeded system development tasks. Users had changing expectation towards the system.	Ensure user's commitment and deal with users' changing expectation. Demonstrate savings from using the package.
Actions taken	IS project should be managed by internal team.	A consensus-building exercise was organized to enhance collaboration between IS staff and product managers. A strategic vision was developed.	A companywide groupware project was introduced. The project office forced an implementation of new system, with the hope to overcome users' inertia.	The IS project office invested in new hardware and software in the hope to improve system development and win over users' confidence.	Adopt a package enterprise system through a switch-over strategy. The IS project office hired external consultants to smooth the conflicts. To demonstrate saving, they reduce front-line operators.
Outcome	Under stress, the IS project was quickly handed over to internal IS function.	System development was delayed. IS staff isolated themselves for technical design.	The groupware was left unused. Users began to exploit local applications to handle current tasks and deal with the project office's IS audit.	Supply chain problems escalated and many product managers left the company.	Data integrity issue; The misfit of context (different regulations and routines) caused more user complaints.

Table 4. Learning Failures in Information Systems Implementation

Organizational learning barriers	Reasons of learning dysfunctions (as observed in this study)	Practical considerations in breaking the cycles of failures
Limits on organizational intelligence	Information overload High staff turnover Stressful working environment *Implications*: information overloads, loss of organizational memory, and stressful conditions inhibit decision makers' cognitive capacities. Decision makers may make incorrect decisions in IS implementation.	Attend to issues inhibiting cognitive limitations. Focus on finding the root cause (e.g. rationalizing internal process improvement and product strategy before implementing information systems).
Disincentive for learning	The users' distrust towards the implementer The implementer's attribution egoism *Implications*: Social defenses promote learning dysfunctions. The quality of decision may be impaired by users' cover-ups, resistance, distrust, and failure avoidance behaviors, as well as by the implementers' self-justification of failed actions. IS implementation will suffer from political in-fighting.	Remove the penalty mechanism; reward the honest admission of failure. Encourage users to bring up bad news for discussion, and the implementer to face the underlying problems.
Inappropriate organizational design	The implementer's centralization of IS resources *Implications*: The ineffective institutional arrangements promote walled-off effect. Moreover, in trying to centralize IS resources, the implementer may set plans that are too ambitious and impractical, and fail to sustain users' psychological ownership.	Promote mutual collaborations among the executive, implementer and users (e.g. using post-action review to accumulate experience in system implementation). Give users more administrative autonomy. In a decentralized organization, where users have greater political influence (such as FoodCo), the implementer may need to delegate more resources to users.
Education barriers	Lack of user training for transferring *configurations* as well as *assimilation* knowledge *Implications*: The implementer needs to allow users to absorb new changes with the help of sufficient education and training programs. The best practices (assimilation knowledge) embedded in one system (e.g. Oracle) may not be suitable for the practices situated in the recipient company's context (e.g. in consumer good industry in Taiwan's business environment).	Continuously invest resources in user training. Effort should be made towards the transfer of assimilation knowledge (e.g. in areas of process improvement and organizational change initiatives) in addition to configuration knowledge. Assess the congruence between the practice supported by the intended system and the actual practice within the organization. Make the necessary appropriation of technology and improvements to work practice before system implementation.

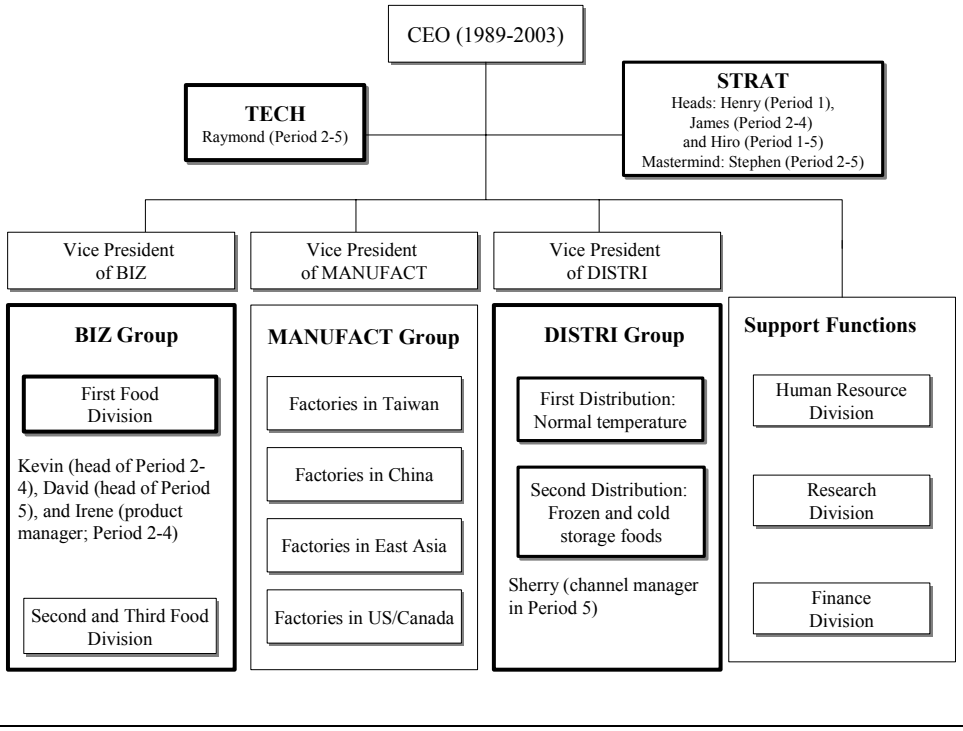

Figure 1. Organizational Chart of FoodCo (Showing the Key Actors)