Pulling the Plug: Software Project Management and the Problem of Project Escalation[1]

By: Mark Keil
Department of Computer Information Systems
College of Business Administration
Georgia State University
P.O. Box 4015
Atlanta, GA 30302-4015
U.S.A.
cismmk@gsusgi2.gsu.edu

Abstract

Information technology (IT) projects can fail for any number of reasons and in some cases can result in considerable financial losses for the organizations that undertake them. One pattern of failure that has been observed but seldom studied is the IT project that seems to take on a life of its own, continuing to absorb valuable resources without reaching its objective. A significant number of these projects will ultimately fail, potentially weakening a firm's competitive position while siphoning off resources that could be spent developing and implementing successful systems. The escalation literature provides a promising theoretical base for explaining this type of IT failure. Using a model of escalation based on the literature, a case study of IT project escalation is discussed and analyzed. The results suggest that escalation is promoted by a combination of project, psychological, social, and organizational factors. The managerial implications of these findings are discussed along with prescriptions for how to avoid the problem of escalation.

Keywords: Software project management, IS failure, escalation, escalating commitment, implementation

ISRL Categories: EE, EE06, EE0101, EE0504, EL0201, EL0202

Some projects never seem to terminate . . . "rather, they become like Moses, condemned to wander till the end of their days without seeing the promised land."

— Stephen P. Keider (1974)

Introduction

By 1994, annual U.S. spending on the development of information technology (IT) applications reached $250 billion (Johnson, 1995). The strategic importance that IT now plays, coupled with the burgeoning costs of developing systems, has raised the stakes associated with project failure. Despite the costs involved, press reports suggest that such failures occur with alarming frequency (Betts, 1992; Cringely, 1994; Ellis, 1994; Gibbs, 1994; Kindel, 1992; Kull, 1986; McPartlin, 1992; Mehler, 1991; Neumann and Hoffman, 1988; Rothfeder, 1988). While it is difficult to obtain statistics on the actual frequency of IT failures, various sources suggest that at least half of all IT projects are not as successful as we would like them to be (Gladden, 1982; Lyytinen and Hirschheim, 1987).

While there are undoubtedly many different modes of IT failure, one pattern of failure that has been observed but seldom studied is the IT project that seems to "take on a life of its own," continuing to absorb valuable resources without ever reaching its objective (Keider, 1974; Lyytinen and Hirschheim, 1987; Meredith, 1988). Eventually, these projects are abandoned (or significantly redirected), but the cost of having funded them can represent a tremendous waste of organizational resources. Why are troubled projects allowed to continue for so long before they are ultimately abandoned or brought under control?

Traditional wisdom holds that information systems projects get "out of control" because of poor project management practices. It would be hard to argue otherwise. But what is meant by "poor project management?" This phrase has become a dumping ground for explanations of IT failure that range from a chronic tendency to underestimate the cost or scope of an IT project (Boehm, 1981; Brooks, 1975; Kemerer, 1987) to failure in managing the risks associated with IT projects (Alter and Ginzberg, 1978; Ginzberg, 1981; McFarlan, 1981). While there is merit behind these traditional views, they do not explain why projects that get out of control seem to stay that way.

Many IT projects that seem to take on a life of their own represent what can be described as escalation. Escalation has been defined as continued commitment in the face of negative information about prior resource allocations coupled with "uncertainty surrounding the likelihood of goal attainment" (Brockner, 1992). Project escalation can therefore be said to occur when there is *continued commitment* ***and*** *negative information.*[2]

In order to both explain this phenomenon and prevent its occurrence, it is necessary to look beyond traditional explanations of poor project management and to consider possible psychological, social, and organizational factors that may promote project escalation.

Background

One of the most difficult management issues that can arise in connection with IT projects is deciding whether to abandon or continue a project that is in trouble. Unfortunately, there is very little information available on the subject of IT project abandonment. One study found that 35 percent of these projects were not abandoned until the implementation stage of the life cycle (Ewusi-Mensah and Przasnyski, 1991). This suggests that IT managers are doing a poor job of identifying or terminating projects that are likely to fail. While there may be several reasons why such a high fraction of abandoned projects are not terminated earlier in the life cycle, one explanation may be that managers have a natural tendency toward escalation or continued commitment to a failing course of action (Brockner, 1992).

Factors that can promote escalation

Previous research suggests that escalation is a complex phenomenon that may be influenced by many different factors. Based on a review of the literature, Staw and Ross (1987a) provide a useful taxonomy that groups these factors into four categories: project factors, psychological factors, social factors, and organizational factors.[3] Using this typology as a basis, Figure 1 represents a model of project escalation.

Project factors are the objective features of the project itself and how it is perceived by management (Ross and Staw, 1993). These factors include the costs and benefits associated with the project as well as the expected difficulty and duration of the project. Other things being equal, projects are more prone to escalation when they involve a large potential payoff, when they are viewed as requiring a long-term

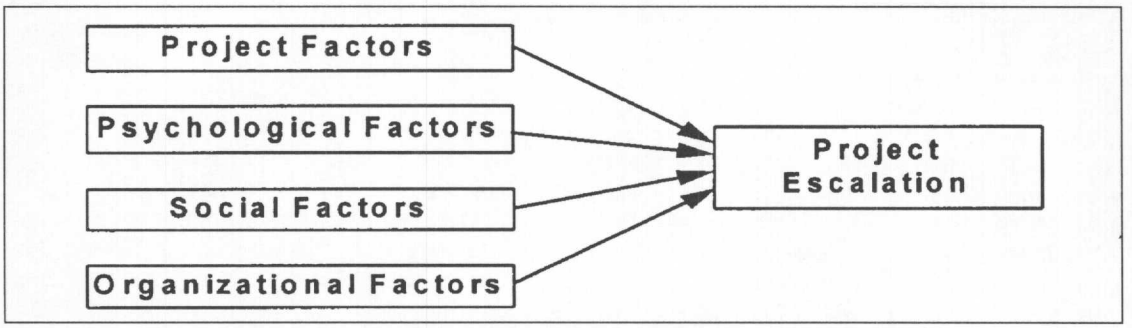

Figure 1. Model of Project Escalation

investment in order to receive any substantial gain, and when setbacks are perceived as temporary problems that can be overcome.

Psychological factors are those that cause managers to convince themselves that things do not look so bad and that continuation will eventually lead to success (Brockner, 1992). These factors include the manager's previous experience with similar projects, the degree to which the manager feels personally responsible for the outcome of the project, as well as psychological and cognitive biases that can affect the way in which information concerning the project is perceived or processed. Projects are more prone to escalation when there is a previous history of success and when there is a high level of personal responsibility.

Escalation is also more likely to occur when managers make errors in processing information. Previous research shows, for example, that human decision making is subject to numerous biases, many of which operate at a subconscious level (Kahneman and Tversky, 1982). One such bias can lead to "throwing good money after bad" in an effort to turn around a failing project (Garland, 1990). Prior research also suggests that managers may engage in a type of self-justification behavior in which they commit additional resources in order to turn a project around rather than terminating the project and admitting that their earlier decisions were incorrect. Self-justification can lead managers to "bias facts in the direction of previously accepted beliefs and preferences," resulting in project escalation (Ross and Staw, 1993, p. 716).

Social factors can also promote escalation. These factors include competitive rivalry with other social groups, the need for external justification, and norms for consistency (Ross and Staw, 1993). Projects are more prone to escalation when competitive rivalry exists between the decision-making group and another social group, when external stakeholders have been led to believe that the project is (or will be) successful, and when norms of behavior favor "staying the course."

Finally, organizational factors involve the structural and political environment surrounding a project. These factors include political support for the project and the degree to which the project becomes institutionalized with the goals and values of the organization. Projects are more prone to escalation when there is strong political support at the senior management level and when the project has become institutionalized.

Prior research on escalation

Prior research on escalation has been based almost exclusively on laboratory experiments that have focused on individual decision making. While these studies have generated useful information concerning factors that may promote or impede escalation at an individual level, there is a growing recognition of the need for more field-based studies that capture the organizational dynamics of the phenomenon (Garland, et al., 1990; Ross and Staw, 1993). To date, there have been only a handful of field-based studies of escalation (Newman and Sabherwal, 1994; Ross and Staw, 1986; 1993). For these reasons, this research focuses on three research questions: (1) Does escalation occur in actual IT projects? (2) What are the factors that seem to promote escalation? and (3) What are the course of events that can break a cycle of escalation?

Methods

Since the objectives of this research were to determine whether the escalation phenomenon could be observed in an actual IT project and, if so, to understand more about the reasons *why* it occurs, this research employed a longitudinal case study approach. Given the exploratory nature of the research, an in-depth case study of a single project named CONFIG[4] was conducted in a company called CompuSys.[5]

Three different kinds of data were collected: interview data (obtained from talking with users, developers, and managers), observations (recorded from meetings that took place involving users, developers, and managers),

Table 1. Scope of the Research

Number of Interviews Conducted[6]	197
Number of Individuals Interviewed	111
Number of Meetings Observed[7]	19
Number of Documents collected[8]	> 350

and historical documents (in the form of meeting minutes, memos, and reports concerning CONFIG). Table 1 gives some indication of the scope of the research.

Table 2 provides an indication of the number of individuals who were interviewed by job function.

Figure 2 presents a schematic diagram of the research design, showing the chronology of the CONFIG project, along with separate timelines showing when various types of data were generated or collected in relation to the history of the project.

Appendix B provides additional information on the methodology and discusses some of the limitations associated with the approach taken in this study.[9]

Table 2. Number of Individuals Interviewed By Job Function

Job Function	#
Sales Representative	48
Sales Manager	7
Developer	6
Development Manager	7
User/developer Liaison	20
Senior Managers	8
Other	15
TOTAL	111

A Brief History of the CONFIG Project and Why It Failed

In the early 1980s, CompuSys, a large computer company, began developing an expert system called CONFIG. CONFIG was designed to help the company's sales representatives produce error-free configurations prior to price quoting. The system was intended to reduce costly "allowances" in which the company had to supply hardware without charge to customers when price quotations were found to have been based on inaccurate product specifications. In addition to the tangible costs associated with "allowances," configuration errors caused several more significant, but less tangible, prob-

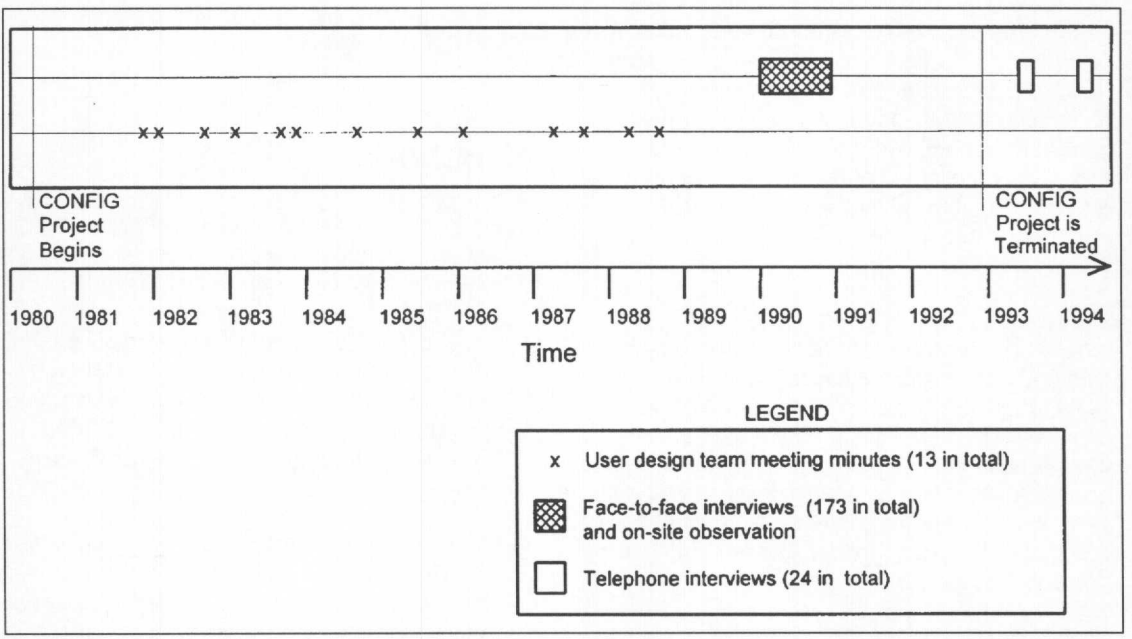

Figure 2. Schematic Diagram of the Research Design

lems for CompuSys. These problems included delays in the order fulfillment process and disgruntled customers who threatened the reputation of the company.

From the very beginning, the CONFIG project faced serious challenges including a lack of sponsorship within the Sales organization, problems of software accuracy and performance, and an unrealistic project schedule. Later, many implementation difficulties were experienced as the system was deployed within the Sales organization. As a result, very few sales representatives actually used the system. While numerous attempts were made to improve the system, the level of usage among sales representatives remained disappointingly low. Nevertheless, the CONFIG project was funded for a period of more than a decade during which it was continually adapted and refined, resulting in a series of redeployments to the field. During this period, both the size and composition of the project team varied, but the commitment to the project was unwavering. The last major effort to adapt and redeploy CONFIG took place during 1989–1990. Despite a long and troubled history, the CONFIG project was continually funded until the end of 1992 when financial support was withdrawn and all further development and support for the project was terminated. A brief description of the project's history and why it failed is provided below.

The business problem and the origins of CONFIG

Throughout its history, CompuSys had prided itself on the almost limitless number of different configurations it offered to customers. This required putting together, or configuring, a group of components that were compatible with one another and that, when combined, would result in a complete and functioning system for the customer. By the 1970s, the growing size and complexity of CompuSys' product line made it increasingly difficult to insure that systems were properly configured when they left the factory. Despite manual efforts to verify that all the necessary components were present, incorrectly configured systems would frequently be shipped to the customer.

In 1975, CompuSys established a task force to study the problem and develop better solutions for verifying system configurations before shipping the goods. In 1978, CompuSys funded the development of an expert system known as VERIFIER that would be capable of verifying and correcting system configurations. As VERIFIER moved through the proof of concept stage, Tom Jones, who later became manager of the company's Artificial Intelligence Technology Center, was recognized as its champion. During the early 1980s, Jones recruited a growing number of individuals to help develop and maintain VERIFIER and to explore other business applications of expert systems. One of the key individuals he brought into the group was George Smith, who was appointed program manager for VERIFIER in 1982 and was heavily involved in managing its transition from a prototype into an evolving production system.

By the early 1980s, VERIFIER was in production use and reportedly saving the company millions of dollars per year, with performance that exceeded some of the company's best human configurers. As a result of both this project and his previous contributions, Jones had secured a high level of status and respect within CompuSys. Having gained some confidence in building VERIFIER, Jones and his group of managers and developers began to search for other opportunities where they could apply the technology. As shown in Figure 3, this was the beginning of a cascading sequence of events that set the stage for project failure in the case of CONFIG.

Prior history of success prompts development of CONFIG

The very success of VERIFIER, which was developed and funded by the Manufacturing organization, led to the realization that the configuration problem originated in the sales offices and that, while VERIFIER helped Manufacturing, it did nothing to assist Sales.[10] With this

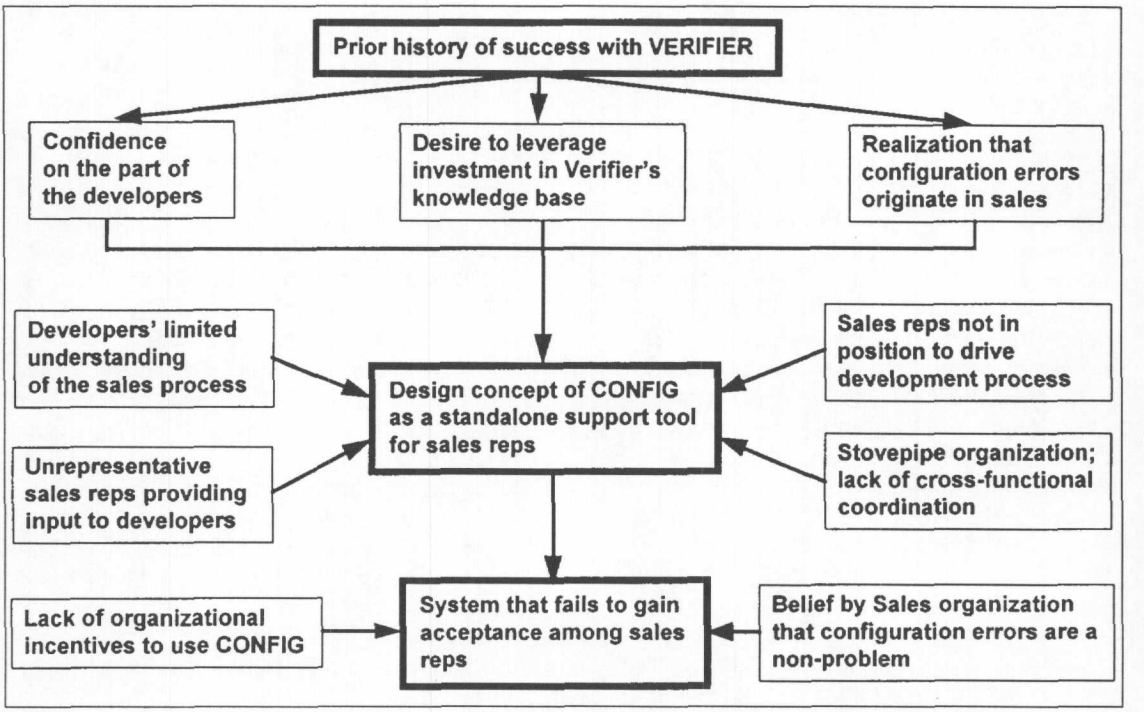

Figure 3. Events That Set the Stage for Project Failure in the Case of CONFIG

realization, the idea for CONFIG was born. Unlike VERIFIER, which was a *batch-oriented tool* used by Manufacturing for configuration *validation*, CONFIG was to be an *interactive tool* that would aid sales representatives in *creating and validating* configurations prior to quotation.

By 1981, the group that developed VERIFIER had begun funding the initial prototype of CONFIG. The CONFIG program manager formed a "user design team" (UDT) composed of about a dozen representatives from the Sales organization. This group typically met twice annually with the developers; its charter was to provide guidance and feedback as additional features and functions were added to the evolving prototype. The UDT met with the developers on 13 occasions between November 1981, and August 1988, when the group ceased to meet regularly.

Initially, there were only a few developers who were assigned to work on CONFIG, but the resources devoted to the project increased substantially during the 1980s.[11] Although the composition of the project team assigned to CONFIG changed over time, both Jones and Smith maintained a high level of involvement.[12] The development activity was funded primarily out of Manufacturing, where the group was housed until the late 1980s. Figure 4 represents a partial organization chart for CompuSys, showing the structure as it existed in 1990 and some of the individuals who were interviewed as part of this study.

A flawed design concept

CONFIG's original design concept called for a standalone support tool for sales representatives. As the project unfolded, the need for integration between CONFIG and the company's quoting system became increasingly apparent, and the standalone design concept proved to be fatally flawed. Ironically, at the same time that the CONFIG project was getting underway, another group of developers that was more closely aligned with the Sales organization was just beginning to develop a new computerized price quotation system (PQS).

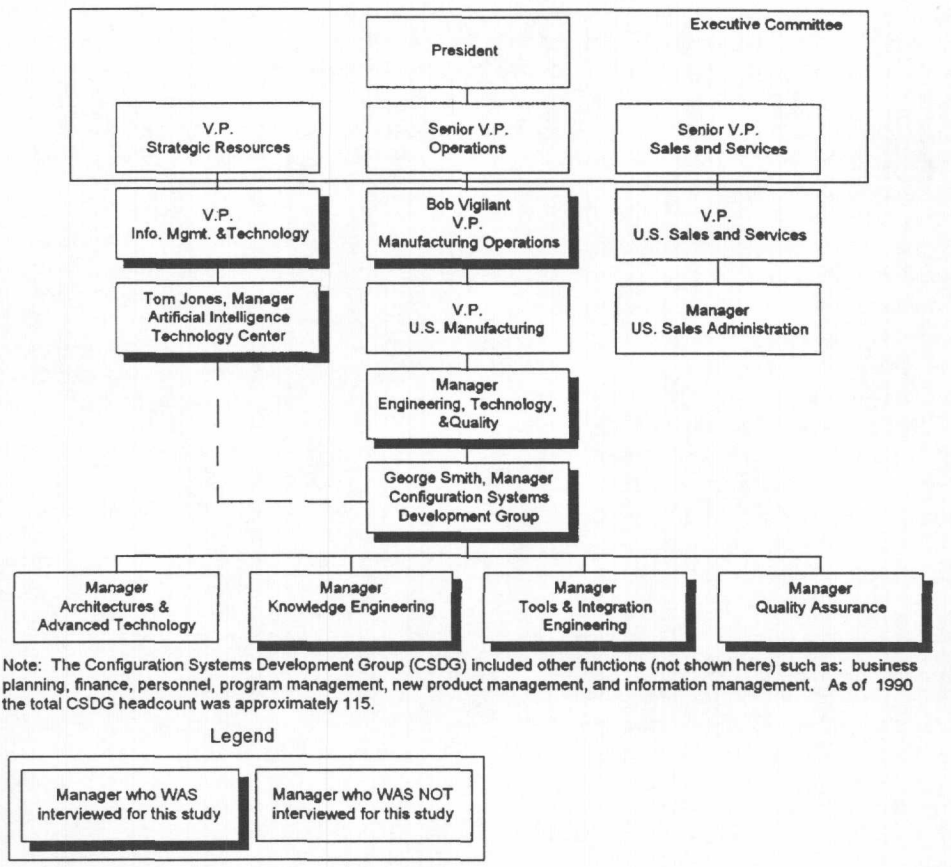

Figure 4. Partial Organization Chart for CompuSys, 1990

While the connection between the configuration process and the price quotation process seems obvious now, at the time, these two systems were being developed and managed as entirely separate processes. This was largely due to a company culture that promoted the creation of "stovepipe organizations," resulting in a lack of cross-functional coordination. As a result, two standalone systems that did not talk to one another were created—one for generating configurations and one for generating price quotations.

The basic design concept of CONFIG—as a standalone support tool for sales representatives—emerged and remained intact for several reasons. First, because the development group resided in Manufacturing, they had a limited understanding of the sales function. Their model of the sales process was one in which configuring and quoting were seen as two separate processes (Keil, et al., 1995a; Markus and Keil, 1994); this model led them to believe that it was not necessary to have accurate price information during the configuration process. The sales force did not operate this way, however; most representatives found that pricing information often drove the configuration process. Thus, it was necessary to work back and forth between the system's technical configuration and what the customer was willing to pay.

As early as 1982, some of the more vocal sales representatives on the UDT had begun to argue that CONFIG and PQS had to be integrated. The developers, however, contended that this would be too challenging technically and made integration a "non-goal" for CONFIG's first release. Since the UDT served only in an advi-

sory capacity, the sales representatives were not in a position to drive the development process. Consequently, the integration of CONFIG and PQS remained a low priority until 1984 when it became more obvious that there would have to be at least some level of integration between these two systems.

By that time, however, the composition of the UDT had shifted and the group was no longer representative of the salesforce at large; as more novice sales representatives lost interest in the process and left the group, the UDT had become dominated by a relatively small number of experienced, technically oriented sales representatives. As a group, they tended to be more forgiving of the software and were willing to compromise on such items as "ease of use" in exchange for a system that was accurate and functional. Rather than arguing for complete integration between CONFIG and PQS, the UDT was willing to settle for a non-interactive one-way linkage. The establishment of such a crude linkage was predicated on a model of the sales process in which CONFIG was still seen as a viable standalone system so long as its output could be passed on to PQS.

While this linkage was made operational in 1985, it was not enough to drive significant usage of CONFIG. In fact, those who used the system found it more time consuming to produce price quotes than those who did not. In the words of one sales representative:

> When you get done with CONFIG and want to print a quote, you create a file, transfer a file, you go into PQS, you import it. It's very cumbersome. The turnaround time is way too long, so I've started not to use the system [CONFIG] because when I need to do a quote, generally the customer is waiting. They've asked for something, and you can't wait a day or two to get stuff back.

System fails to gain acceptance among sales reps

In addition to the fact that the tool was based on a flawed design concept, there were other reasons why CONFIG failed to gain acceptance among sales representatives. One reason was a lack of organizational incentives to use CONFIG. Quite simply, sales reps were not motivated to produce error-free configurations; they were rewarded on the basis of sales volume, not configuration accuracy. In the words of one sales rep:

> It's not that anyone wants to be inaccurate and make a lot of errors, it's just not something we are measured on, and we will work toward what we are measured on.

A sales manager who was asked to explain how reps' performance was evaluated said:

> You won't see anything [in the performance appraisal form] about quote accuracy. I'm not managed, nor is my manager managed, to "dirty" orders. So it doesn't really matter. It's not one of my metrics. It's not a critical success factor for me or for the sales reps. There are no kudos if they get it right, no scolding if they get it wrong.

Under this type of reward system, the costs associated with inaccurate configurations were never factored into the sales reps' compensation. Therefore, most of the people in sales (from district managers on down) thought that the configuration error problem was too minor to justify the effort of using CONFIG (Markus and Keil, 1994):

> We never miss the big stuff . . . CPU, etc. . . . only cables. Who cares about a $50 cable? It's not worth the time it takes to run it through CONFIG. Ninety percent of the time, the customer will authorize a modification anyway.

Was CONFIG an Example of Project Escalation?

As stated before, escalation can be defined as *continued commitment* of resources in the face of *negative information* (Brockner, 1992). In order to investigate whether these conditions actually existed, a detailed chronology of the project was reconstructed using the UDT meeting minutes and, in some cases, actual transcripts of UDT meetings as a foundation. The 13 UDT meetings that took place during the life

of the project, along with the decisions to initiate and terminate the project, were chosen as natural decision points for analysis.[13] Where possible, the information from the UDT meeting minutes was supplemented by examination of agenda items, priority lists, and presentation slides associated with each meeting. Interview and historical data were then used to reconstruct events that occurred between UDT meetings. The resulting chronology was then validated by several individuals who were familiar with the project's history.

Using this chronology of events, two independent raters were then asked to assess the character of the information that was available to decision makers during the course of the CONFIG project. Based on their assessment, project information was coded as positive, negative, or ambiguous. Excerpts of the project's chronology and the raters' coding of project information can be found in Appendix A.[14] As shown in Figure 5, the CONFIG project continued for a period of more than a decade despite the fact that information concerning the project was predominantly negative during this period of time. Therefore, the CONFIG project satisfies the definition of project escalation.

Objective usage data as well as the subjective opinions of individuals closely associated with the project offer confirmatory evidence of escalation. Figure 6, which shows CONFIG usage as a percent of system quotations, reveals a pattern of declining use during the last several years of the project's history.[16]

The fact that CONFIG was used for a very small percentage of system quotations and that its usage declined steadily during a four-year period represents significant negative information concerning the project. This information was readily available to decision makers within the company.

Qualitative interview data gathered during 1990 provides additional evidence. As one manager observed: "I think the thing may have taken on a life of its own." By 1990, many sales representatives who were interviewed expressed a belief that the CONFIG project should be abandoned. The following remarks were typical:

> Based on the usage patterns, I don't think anybody would miss [CONFIG] very much if we turned it off tomorrow.
>
> If [CompuSys] has spent millions on this product we have really missed [the boat]. [We should] pull the plug on this effort immediately.
>
> The people responsible for developing [CONFIG] are trying to breathe life into something that should be allowed to die . . . We have proof today that [CONFIG] is not successful. It has failed miserably. The problem is nobody is willing to kill it.

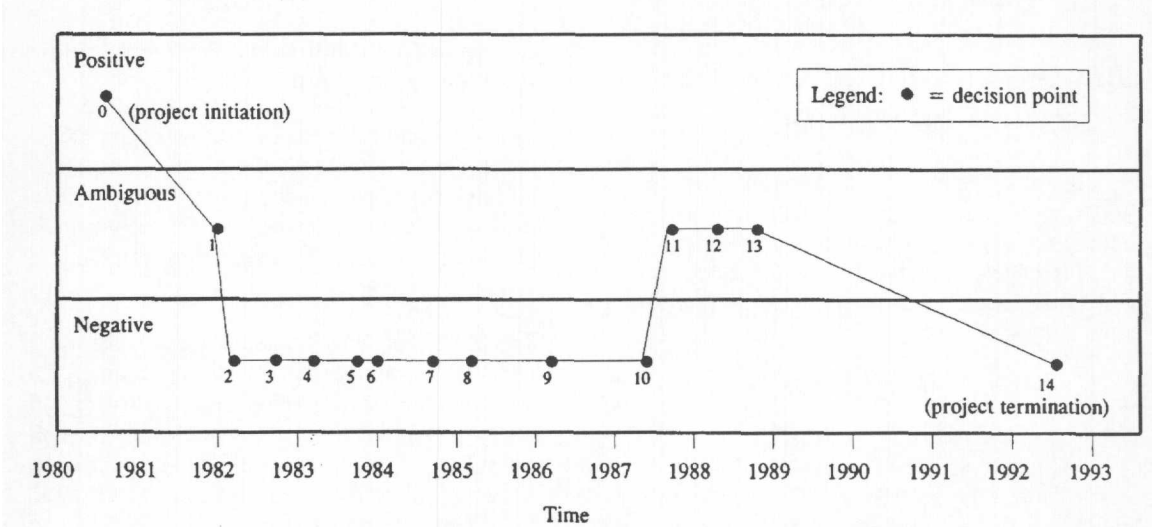

Figure 5. Mapping CONFIG Project Information[15]

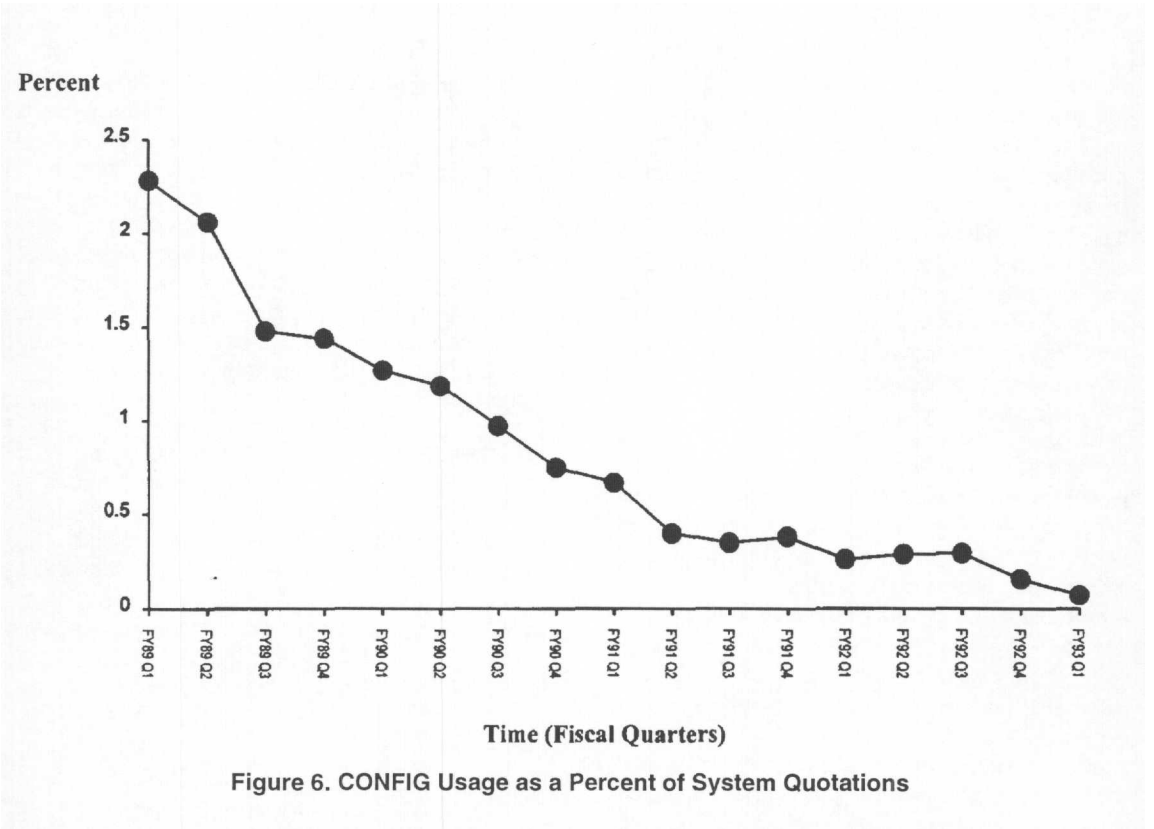

Figure 6. CONFIG Usage as a Percent of System Quotations

Given what appears to be such a clear pattern of negative information one would have expected the project to be terminated or redirected much earlier than 1992, the year in which the project was finally halted. How, then, can the escalation that occurred be explained?

Discussion of Factors That Promoted Escalation

As suggested by the model presented earlier (see Figure 1), there were four different types of factors that seemed to contribute to the escalation of the CONFIG project: project factors, psychological factors, social factors, and organizational factors.

Project factors

Some factors associated with the characteristics of the project and how it was viewed by management were: (1) there was evidence that continued investment could produce a large payoff, (2) the project was regarded as an investment in research and development, and (3) project setbacks appeared to be temporary problems.

Evidence That Continued Investment Could Produce Large Payoff

Three separate financial analyses of the CONFIG project conducted in 1982, 1985, and again in 1987 provided evidence that a successfully developed and deployed system could have yielded a positive and significant net present value for the company.

The 1982 analysis indicated a net present value of $43.9 million (20 percent discount factor) for the five-year period FY82–FY86. Operating and development costs for the project were projected at $10.4 million between FY82 and FY86. A second financial analysis conducted in 1985 projected a net present value of $55.7 million (20 percent discount factor) for the five-year

period FY85–FY89. A third financial analysis conducted in 1987 indicated a net present value of at least $41.1 million (20 percent discount factor) for the five-year period FY87–FY91.

In each of the analyses, it was argued that under virtually all scenarios, CONFIG had a very large potential value. Since all three financial analyses revealed a positive and significant NPV, the potential payoff associated with successful completion of the project may have served as a strong justification for continuation. It is important to note that there were other intangible benefits that were thought to be attainable as well and that these may have had an effect on escalation that was equal or greater than the tangible economic benefits that could be obtained.

Project Regarded as an Investment in Research and Development

Within CompuSys, there was a strong belief that artificial intelligence held great promise as a technology that could be used to solve complicated problems both within the company and for the company's customers as well. One manager who was closely associated with the project explained the escalation in these terms:

> I think it was a combination of optimism which you could call undue or not and a sense that this was a new technology that we were applying to this problem and that experimenting with it could yield the results that we wanted even if we couldn't see them in front of us at the moment. So there was a kind of technological optimism . . .
>
> There was always a sense that [Jones] would find a way to keep the project going. I think that [Jones] took for granted that the problem was real and that the technology held promise to solve it. He never wavered on that. Even though we sometimes got a cold reception from Sales, it continued to be funded anyway as an R&D project.

In the words of one executive:

> Senior management had developed so much faith in the technology that they continued to fund it regardless of what the data said.

Project Setbacks That Appeared to be Temporary Problems

The limited acceptance among sales representatives was always regarded as a temporary problem that could be overcome with additional resources. Several user surveys that were conducted (in 1984, 1985, and 1986) seemed to suggest that the sales representatives did have need for better configuration support tools in the field and that they might use CONFIG more if issues such as lack of availability, lack of training, poor ease-of-use, and poor response time were resolved (CompuSys internal memorandum, 1987).

To the development team, this feedback implied that the tool could be made acceptable to the sales force with some additional development work and more emphasis on providing the availability, training, and support needed for a successful implementation. This led not only to the continuation of the project but to a massive effort in 1989 to improve the tool's user interface (Keil, et al., 1995a) and to redeploy it with "the necessary infrastructure to ensure optimization of the configuration creation process" (CompuSys internal memorandum, 1989). But despite introducing a far more usable tool with a near textbook implementation program, this multimillion-dollar redeployment effort failed to have a significant impact on sales reps' willingness to use CONFIG (Keil, et al., 1995a; Markus and Keil, 1994).

Psychological factors

Some psychological factors apparently caused the key decision makers, Jones and Smith, to convince themselves that continuation would eventually lead to success (Brockner, 1992). These factors included the following: (1) prior history of success, (2) high degree of personal responsibility for the outcome of the project, (3) errors in information processing, and (4) emotional attachment to the project.

Prior History of Success

The success or failure of previous projects is a psychological factor that can influence a man-

ager's decision frame for identifying, interpreting, and acting on project information. Prior history of success was evident in the case of CONFIG. The prior success of VERIFIER naturally caused the managers responsible for CONFIG [Jones and Smith] to be confident about its prospects for success. As one individual who was close to the project observed:

> What I saw were [Jones and Smith] building upon a previous success with [VERIFIER] and saying we can hit that home run again.

Staw and Ross' (1987a) model of the psychological determinants of commitment suggests that a prior success may inhibit a decision maker's willingness to re-examine the current course of action, thus promoting escalation. Psychological literature on selective perception (e.g., Allen and Marquis, 1964; Hastorf and Cantril, 1954; Hogarth, 1979) provides additional support for this notion. One would expect decision makers to ignore negative information or to downplay its significance when there is a prior history of success.

High Degree of Personal Responsibility for the Outcome of the Project

Based on interviews conducted with a wide range of individuals throughout the company, it was clear that Jones and Smith were perceived to be the managers most responsible for the continuation of the CONFIG project. Historical data gleaned from company memos and reports confirmed that both Jones and Smith were closely associated with the CONFIG project for a decade or more. Furthermore, many interviewees were quick to point out that CONFIG was Jones' "baby," implying that he was the individual responsible for having initiated the project.

According to escalation theory, individuals with a high degree of personal responsibility will have a tendency to engage in self-justification; they will convince themselves that it is better to continue than abandon because they want to show that their original decision to pursue the project was "correct." The need to self-justify is heightened when prior expenditures are irrevocable, public, freely chosen, and repeated—all conditions that existed in the case of CONFIG. The net effect of self-justification is to lower the probability that questionable projects will be reconsidered.

Errors in Information Processing

Managers who reported to Jones and Smith suggested that neither one of these two individuals wanted to hear negative information. Instead, Jones and Smith apparently sought out positive information about the project. As one manager who was closely involved recalled:

> I don't think they [Jones and Smith] ever perceived any problem with the product. If they did, they never shared it with me or as far as I know anybody in the group. They certainly never wanted to hear that there was any problem.

Another individual explained how Jones and Smith would react when presented with negative information:

> When we did surveys of people in the field the information we got wasn't what they [Jones and Smith] wanted, but you know denial is so powerful . . . Their response was: "Wrong answer, we don't like that answer." They never listened to [negative] feedback.
>
> When I approached [Smith] with negative feedback from the field, he would get defensive about it. He would always say: "Well, that's hearsay." And we would have to go out and conduct additional surveys.
>
> He seemed to be exercising denial. [He'd say] "There must be something wrong with your survey process. Talk to these people individually." Sometimes we would be asked to go back and survey the same individuals again . . . and even if the information was still negative, [Jones and Smith] would find some way to put a positive spin on it.

The type of behavior described above is consistent with escalation theory. As Staw and Ross (1987a, p. 54) observe: "If the objective facts disconfirm one's opinion, the individual will work hard to find reasons to discredit the source of information or the quality of the data itself."

Emotional Attachment to the Project

Several key members of the development team expressed a strong bond with the project. In the words of one manager:

> I'm trying to be very objective and pull myself away from being *so attached* to [CONFIG]. After you've had blood, sweat, and tears over it for the last three years, *you get very attached* to [CONFIG] and very defensive of it (emphasis added).

Another individual who had been one of the original CONFIG developers explained his reluctance to abandon CONFIG:

> I don't think we want to get rid of [CONFIG]. I don't want to just throw out 10 years of work . . .

Given the level of emotional attachment displayed by various members of the development organization, it is reasonable to assume that Jones and Smith were also emotionally attached to the project. Several sales managers who were in a position to observe the development process commented on the emotional attachment that existed on the part of the development organization in general. As one manager observed:

> Some of these guys had five or six years of their life in that project. Their life was in that . . . the emotional baggage of hanging on to it . . . not being able to say: "this isn't going to work" and walking away from it . . .

Social factors

There also appeared to be some social factors that contributed to escalation: (1) competitive rivalry between Sales and Manufacturing, (2) need for external justification, and (3) norms for consistency.

Competitive Rivalry Between Sales and Manufacturing

Within CompuSys, the competitive rivalry that existed between Sales and Manufacturing was legendary. It is in the context of such rivalry that CONFIG's developers never seemed to consider that their basic design concept might be flawed. While many in Sales viewed CONFIG as a "turkey" of a system, Jones and Smith repeatedly attempted to blame the Sales organization for CONFIG's low acceptance. As two different managers observed:

> They [Jones and Smith] kept saying: "the sales force doesn't understand the tool. They don't want to take the time to learn it."

> The spin that they [Jones and Smith] would always put on the project was that the field just didn't understand . . . that the field just didn't support it.

As research on self-serving biases has shown, individuals are much more likely to attribute negative outcomes to external rather than internal causes (Staw and Ross, 1978). In the case of CONFIG, there is considerable evidence that decision makers saw the failing outcome as the result of ignorant or obstructionist behavior on the part of the sales force. Seen in this light, continuation of the project may have been viewed as the only way for Manufacturing to justify its investment and at the same time exert its will over Sales. For Manufacturing, the alternative of terminating the project would have been tantamount to admitting that Sales was "right" and that manufacturing was "wrong."

Need for External Justification

The need to justify a course of action can be heightened when a decision maker seeks to rationalize his/her behavior to other parties. In the case of CONFIG, there were external constituencies—such as customers and shareholders—that were led to believe that CompuSys was successfully pursuing artificial intelligence both to improve internal processes and as the basis for new service offerings to its customer base.

As early as 1982, journal articles and books began to appear describing the efforts that were underway at CompuSys to build expert systems such as CONFIG. In these writings, CONFIG was typically portrayed as a successful example of how CompuSys had embraced

new technology to develop a leading-edge system that would improve customer service. Ironically, there were some sales representatives who never embraced the tool but saw the value in it from a public relations standpoint and referred to it in discussions with customers.

Norms for Consistency

Norms for consistency represent another social variable that may have promoted escalation. In the U.S., for example, substantial rewards are often given to managers "who can turn things around or convert a losing project into a winner" (Staw and Ross, 1987a, p. 58). Prior research has shown that managers who exhibit consistent commitment to a course of action are perceived as strong leaders, suggesting that "persistence is a socially valued style of leadership" (Staw and Ross, 1987a, p. 59). This combination of social norms suggests that there may be a kind of "hero effect" in which society reserves special praise and admiration for leaders who "stick to their guns" and are able to turn things around even when there is a low likelihood of success.

In addition to societal norms that may have favored persistence, there were also strong social norms *within* CompuSys that favored continuation. During the 1980s, it was not part of the company culture at CompuSys to kill projects; instead, failing projects were allowed to die from natural causes. In the words of one manager:

> Part of the culture then was that it was okay to let things linger before someone actually went out and killed them. That was not unusual.

In the case of CONFIG, this meant that managers were unwilling to terminate the project. As another manager observed:

> Nobody wanted to kill it. We're a humanitarian organization. Nobody wanted to shoot it in the head.

Organizational factors

The fourth and final set of factors that may have promoted escalation were organizational (or political) in nature and included the following: (1) strong advocates who provided continued funding and protection, (2) empire building, and (3) slack resources and loose management controls.

Strong Champions Who Provided Continued Funding and Protection

According to one senior manager:

> There were always signals that CONFIG wasn't being embraced [by the field], but there were some very strong champions inside Manufacturing—largely [Vigilant] and [Jones]. [Vigilant, who was the Manufacturing VP] kept funding it out of his discretionary funds . . . This was a classic example of how [CompuSys] would continue to invest in a project when there were strong advocates—regardless of whether or not it made financial or market sense to do so.

Because of these strong advocates and their high position within the organization, other senior managers within Manufacturing who were critical of CONFIG had only a weak voice. In the words of one senior manager in Manufacturing:

> As long as [Vigilant] and [Jones] had control over it, there was nothing that I or anyone else could do.

While some senior managers outside of Manufacturing argued that CONFIG should be terminated, these arguments seem to have fallen on deaf ears. As one manager recalled:

> I wrote three whistle-blowing memos to three vice presidents . . . I couldn't stop it. It was a sacred cow project.

The presence of such strong advocates clearly promoted escalation. This is consistent with Staw's observation, "If advocates of a project are represented on governing bodies and budget committees charged with the fate of a venture, one may expect substantial persistence in the course of action" (Staw and Ross, 1987a, p. 61).

Empire Building

During the 1980s, Jones and Smith had created and staffed an entire organization whose existence was at least partially dependent on the CONFIG project. Many felt that Jones and Smith had a disincentive to abandon CONFIG because such an action would have threatened the growth of their "empire," thus diminishing their status. The following remarks from three different individuals were typical:

> They [Jones and Smith] were always trying to grow. The staffing level for [CONFIG] never went down. It only went up . . . [CompuSys was] in a period of relative growth and building empires was the thing to do because that was an indicator of how powerful you were . . . It was like: 'Hey, we're going to have our own building.'

> It was kingdom building—people preserving their kingdom and building their power base in order to sustain what they wanted to do in order to make themselves important.

> CompuSys is real good at creating fiefdoms. It's pretty typical throughout the corporation. And I think this was a classic case.

A reasonable interpretation of the above evidence would suggest that Jones and Smith may have had reason to believe that terminating the CONFIG project would diminish their status within CompuSys.

Slack Resources and Loose Management Controls

During the early 1980s, when CONFIG was being developed, CompuSys had plenty of cash, and project justification mechanisms were loose or non-existent. One manager offered the following explanation of how projects were initiated at CompuSys:

> In the early 1980s you'd sort of go tin-cupping to different parts of the organization. When you got enough money raised you went into development. Hopefully while you were tin-cupping you were still in some sort of needs analysis and design. But at some point you reached a critical mass of support and you built it. Oddly enough, you probably never asked the people who were going to use it if they wanted it . . . you went to people who you thought would want to see this nice thing made. You didn't have to go to any board or council or group or management committee.

In addition to the unusually loose way in which projects were initiated and justified, once a project was started, formal reviews were seldom performed. In the words of one manager who was closely associated with the CONFIG project:

> [CONFIG] was subject to review in a somewhat haphazard way. Sometimes it seemed to me that decisions about this kind of thing were made almost in an off-hand way.

Another senior manager explained how the CONFIG project continued to be funded out of discretionary R&D funds without undergoing any type of formal review process:

> [As along as Tom Jones was in charge] it never got reviewed. It didn't get into the official budget review process until 1992.

The combination of slack resources and loose management controls that existed at CompuSys during the 1980s led to an environment that actually promoted escalation.

Summary model of factors that promoted escalation

Figure 7 represents a summary model of the factors that promoted escalation in the case of CONFIG. Consistent with theory, the escalation appeared to be promoted by a wide variety of variables including project, psychological, social, and organizational factors. While most of these factors have been discussed in the escalation literature, the case study revealed several additional factors that have not been widely discussed in previous studies. These factors include: emotional attachment to the project, empire building, and slack resources and loose management controls.

Reasons for the Eventual Termination of CONFIG

After more than a decade of development and tens of millions of dollars, the CONFIG project was eventually terminated at the end of 1992. The pattern of escalation was ultimately broken by two factors: (1) the passing away of the project's primary champion, and (2) an external shock to the company.

Project's primary champion dies

In 1992, Tom Jones, the project's primary champion, died abruptly of cancer. As one senior manager recalled: "When [Jones] died, there ceased to be enough support to keep it going." Another manager reflected:

> [Jones] could always find a way to pull the money that he needed to keep it alive. Because he believed in the thing. It was the thing he had staked his career on. After [Jones] died he wasn't around to protect it. He wasn't there to influence the funding.

External shock to the company

By the end of the 1980s, CompuSys was confronted with a downturn in the U.S. computer market. Faced with mounting losses during the early 1990s, the company was forced to begin laying off employees and restructuring its operations. Suddenly, a company that was used to operating in a growth mode found itself fighting to regain profitability. This chain of events represented an external shock to the company that appears to have contributed to the cancellation of the CONFIG project. In the words of one manager:

> After [Jones] passed away, there were a lot of organizational changes and [several of] the Manufacturing VPs who had supported [CON-

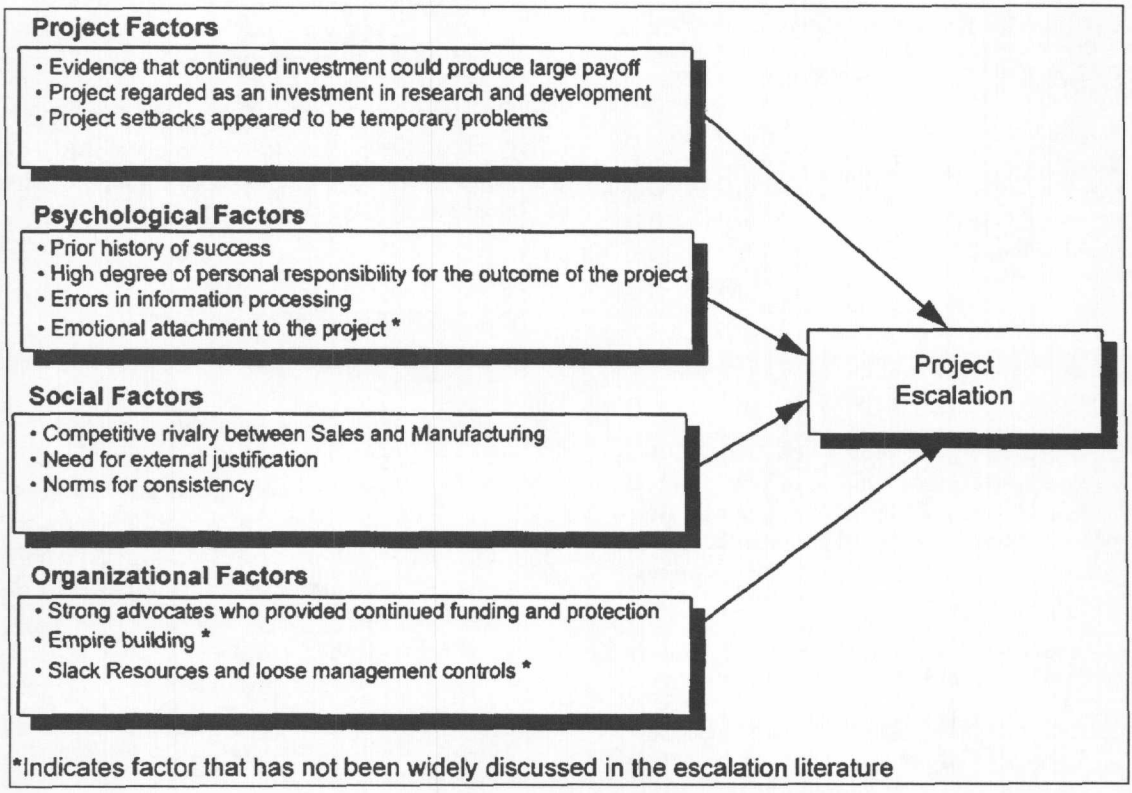

Figure 7. Summary Model of Factors That Promoted Escalation

> FIG] left the company. Then a new guy came in to a situation where resources were scarce because we were in a cost-cutting and downsizing mode. . . He's like: 'What am I doing? We're spending this on something that nobody really wants. Stop.' He wasn't invested personally in it . . .

Other managers offered similar views, suggesting that the external shock to the organization was a contributing factor in the eventual termination of the project. The following remarks from three different individuals were typical:

> I firmly believe that if we had not run into the financial problems that we ran into as a corporation that [CONFIG] would still be alive and well.

> At this point in our history we are killing projects left and right unless they start producing fairly quickly. I think this is due to the cash constraints that we now face . . .

> Management and controls placed on everything are much more stringent now. There are better established program guidelines now. Projects and programs are much tighter defined.

Summary model of organizational exit from escalation

Figure 8 summarizes the events that set the stage for organizational exit in the case of CONFIG.

Consistent with an earlier model (Ross and Staw, 1993), changes in top management were judged to be an important factor in breaking the pattern of escalation. In the case of CONFIG, however, the death of the project's primary champion was also observed to be a key factor in the eventual withdrawal.

Implications

Given that escalation occurs in IT projects, it is important to understand why it occurs and how it can be avoided. Though the findings reported here are based on a single case study, they have significant implications for practice.

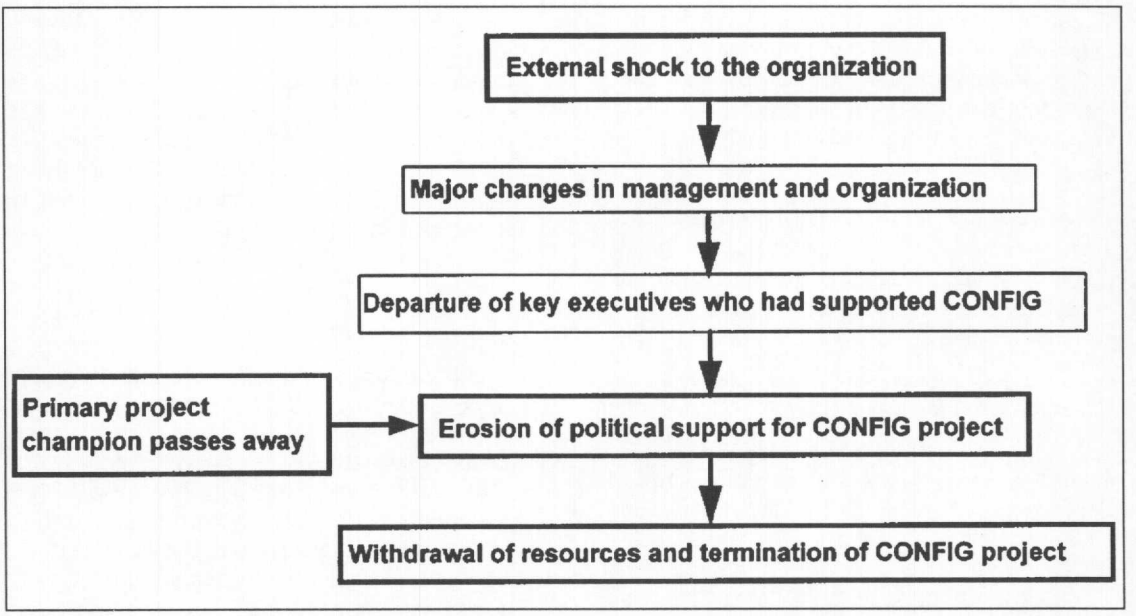

Figure 8. Events that Set the Stage for Organizational Exit

Implications for IT managers

For IT managers, the knowledge that escalation can occur in IT projects underscores the important role that psychological, social, and organizational factors can play in the successful management and control of IT projects. In the past, IT managers have relied upon traditional project management techniques to manage and control IT projects. While the traditional approaches are useful, it should be noted that they are based on a rational approach to project management and thus tend to ignore some of the other dimensions that seem to be associated with project escalation. The research results reported here suggest the need for a broader view of project management and control that encompasses both the rational approach for controlling projects and a more psychological or behavioral perspective.

To avoid the escalation trap, IT managers can take actions as individuals to minimize their own risk of becoming overcommitted, and they can also institute organizational policies and practices to reduce their organization's exposure to escalation. The first step in avoiding IT project escalation is for managers to recognize that there is a natural tendency to escalate when one becomes too committed to a course of action. Although there is often a fine line between "an optimistic, can-do attitude and overcommitment," there are several questions managers can ask themselves to help determine whether they have crossed the line (Staw and Ross, 1987b, p. 72):

- Am I unable to clearly define what would constitute failure for this project? Has my definition of what would constitute success or failure changed as the project has evolved?
- Do I have trouble hearing other people's concerns about the project?
- Am I more concerned about the welfare of this project than I am about the organization as a whole?

If the answer to one or more of these questions is "yes," then the manager is probably too committed to the project and needs to re-evaluate the project before committing any additional resources. Managers can also take steps to avoid becoming overcommitted to a project in the first place. One approach is to periodically evaluate the project from an outsider's perspective. A good question to ask oneself is: "If I took over this job for the first time today and found this project going on, would I support it or get rid of it" (Staw and Ross, 1987b, p. 72)? Another approach is to always consider alternative courses of action in deciding whether to abandon or continue a prior course of action (Keil, et al., 1995b; Northcraft and Neale, 1986). In the case of CONFIG, this approach might have prompted the managers and developers to consider either alternative solutions to the configuration problem or alternative development projects more worthy of resources.

Implications for organizations

While individuals play an important role in the escalation process, "much of what causes escalation is in the nature of organizations, not people" (Staw and Ross, 1987b, p. 72). Without the right structures and incentives it is naive to expect that all managers will be motivated to ask the above questions. There are several steps, however, that organizations can take to create an environment in which managers are forced to raise the kind of questions that can avoid project escalation. The prescriptions that follow are meant to be suggestive of the types of actions that organizations can take, alone or in combination, to help minimize the risk of project escalation.

Know the Stage of the Project and Manage It Accordingly

Organizations should create systems to insure that project *management* is appropriately matched to the stage of the project. Projects that are initiated to explore applications involving new information technologies need to be managed very differently once they are moved from research mode into full-scale development.

One of the problems with the CONFIG project was that it was regarded as an R&D project even after it had been fully developed and implemented. To avoid this problem, companies should create a tracking system that gives senior management a full accounting of all IT projects and their current stages. Clear guidelines should be established to mark the point at which projects move from one stage to another. In the case of CONFIG, this would have prevented its supporters from using available "discretionary funds" to continue their so-called "R&D project" at the company's expense.

Assess Risks Early (and Often) During the Development Process

As in the case of industrial research and development, IT projects typically progress through a number of defined stages (e.g., analysis, design, development, implementation) between initiation and "commercialization." At the earliest possible stages of this process, managers need to begin asking whether there are any "red flags" or serious exposures they will face in continuing to pursue a project.

Unfortunately, for most IT projects, risk assessments are usually conducted on an infrequent and informal basis if they are even conducted at all (Ropponen and Lyytinen, 1993). However, since most organizations follow some type of software development methodology, it would be a relatively easy matter to include a formal and periodic risk assessment as part of the overall methodology for developing systems. While the precise implementation of such an approach will vary from company to company, the two critical areas that should be included in any risk assessment are: (1) the probability of technical success, and (2) the probability of customer acceptance (Balachandra and Raelin, 1980).

In assessing the probability of technical success it is important to consider both the current state of the technology (i.e., how mature it is) and the prior experience of the project team in dealing with the technology (McFarlan, 1981). Given the pace of change in information technology and the constant evolution of new hardware and software platforms, technical success should never be taken for granted. This is particularly true in the case of large, cutting-edge, projects. That being said, the more significant risk often lies in the area of customer acceptance. In assessing this risk, it is important to ask whether there is sufficient "demand" for a software application *before* investing huge sums of money to fully develop it. Like good marketers, IT managers need to conduct market research in the form of focus groups, surveys, and observational studies. While such research is costly, it is necessary to insure that the system design concept is appropriate. As part of the overall evaluation process, IT managers must also consider whether the right organizational incentives exist for people to *want to use* a proposed system. In the case of CONFIG, this type of analysis would have highlighted the exposures that the project team would later face when they found out that the design concept was flawed and that sales representatives had no incentive to use the system. Finding out about these problems early in the process might have avoided project escalation by allowing the project to be terminated or redirected at an earlier stage.

Conduct Serious Project Audits

No development methodology—even one that includes risk assessment—will prevent escalation unless managers are motivated to follow the methodology. Therefore, to provide the proper incentives, every major IT project should be subjected to a periodic audit process led by someone who is appointed to serve as the organization's devil's advocate. This individual should be charged with protecting the interests of the organization as a whole and should not have a stake in the projects that are being audited. The devil's advocate should be someone who is not only experienced in the area of project management, but who has been trained to ask tough questions and to recognize escalation situations. To insure that his/her recommendations will carry clout and to avoid recrimination, the devil's advocate should

report directly to senior management (preferably the CEO).

The devil's advocate would be responsible for assembling a review board. The board should consist of six to eight individuals; some of the members should be selected on a project by project basis, while others should be appointed to serve the board for a fixed number of years. Three or four individuals representing the company's accounting, IS, and human resource management areas should be appointed to serve on the board for three-year terms that could be staggered to provide some continuity. In reviewing projects, this group would then be responsible for including representation from those areas of the business that will be most affected by the system under review.

The review board should normally meet to review projects on a quarterly basis, although the exact review interval will vary with the size and importance of the project. The review itself should be focused on whether or not the project has achieved specific and measurable goals that are tied to system use and business value. Continued support for the project should be withheld if the goals are not specific or if they are specific but are not being achieved. The review should also involve a careful examination of project risks and exposures. Any significant deterioration in the risk profile of the project should trigger a re-examination of the project to determine if support should be withdrawn. In the case of CONFIG, the approach outlined above would have insured that the project was not subject to "haphazard reviews" and might have led decision makers to question their continued commitment to the project at a much earlier stage. In addition, it would have prevented the Manufacturing organization from continuing to fund a sales support tool that the Sales organization did not want and refused to use.

Reduce the Need for Self-Justification

In order to reduce the psychological need for self-justification, organizations can separate initial and subsequent decision-making concerning a particular course of action (Staw and Ross, 1987a). Subsequent funding decisions should not be made by those who have vested interests in justifying the initial course of action. A more radical approach is to rotate managers in and out of projects to insure that those who initiated the project are periodically replaced by individuals who have more objectivity. In the case of CONFIG, either of these techniques would probably have caused the project to be terminated much earlier.

Another way of reducing the need for self-justification is to reduce the consequences associated with failure (Brockner and Rubin, 1985). In many organizations, key mistakes can have an adverse effect on one's career or lead to job termination. Such a climate is likely to heighten the need for self-justification. Organizations that are more tolerant of failure, however, are less likely to experience project escalation (Staw and Ross, 1987a). It is important, however, to create an appropriate balance. An environment of infinite tolerance could prove to be just as prone to escalation.

Conclusion

This research has shown that escalation can occur in an IT project and has highlighted some of the factors that may contribute to the problem. A company that continues to "throw good money after bad" on an IT project is making a very bad business investment for three reasons. First, the infusion of additional dollars is not solving the business problem for which the system was intended. Second, escalation means that the company is continuing to waste valuable resources. Third, there is an opportunity cost because the company is missing the benefits from alternative uses of these resources. Thus, preventing IT project escalation can be critical in determining whether firms are obtaining real value from their investments in information technology (Markus and Keil, 1994). For researchers and practitioners alike, escalation offers a new perspective on software project management that holds the promise of improving our ability to successfully manage IT projects.

Acknowledgements

I would like to thank Jeff Smith, Lynne Markus, Kalle Lyytinen, Eph McLean, Detmar Straub, and Duane Truex, for their helpful comments on earlier drafts of this paper. I would also like to thank the associate editor and four anonymous referees for their constructive feedback. This paper is based on research that was conducted as part of my doctoral dissertation, and I would like to acknowledge the advice and support received from thesis committee members: Jim McKenney, Dorothy Leonard-Barton, John Sviokla, and Benn Konsynski. Research funding from the Harvard Business School's Division of Research is also gratefully acknowledged. Finally, this work would not have been possible without the participation of CompuSys and its employees.

Endnotes

[1] An earlier version of this paper was presented at the 1995 Academy of Management Conference and published in its proceedings (Keil, 1995a). An earlier formulation of some of the ideas presented here also appeared in Keil (1995b).

[2] Escalation does not necessarily imply an increasing rate of investment over time, but rather, refers to a growth in the cumulative amount of resources invested over time. Thus, escalation can be thought of as continued commitment.

[3] Staw and Ross (1987a) initially used the term "structural determinants," which they subsequently relabled as "organizational determinants" (Ross and Staw, 1993).

[4] One of the reasons for choosing this particular project was that the history of CONFIG had been well documented and could be studied by reviewing a variety of historical material. The author is particularly indebted to Dorothy Leonard-Barton for sharing data that she collected on the CONFIG project from 1984–1987.

[5] The company name, project names, and the names of the individuals involved have all been disguised to provide anonymity.

[6] Includes multiple interviews with the same subject.

[7] Includes six conference calls.

[8] Includes 13 user design-team meeting minutes, three cost/benefit analyses, six internal reports concerning barriers to CONFIG usage, and more than 275 problem reports filled out by users.

[9] Since portions of Appendix B make reference to some of the individuals involved in the project, readers who are interested in this material will find it more meaningful to consult the appendix *after* reading about the history of the CONFIG project and why it failed.

[10] By the early 1980s, CompuSys managers had determined that roughly 25 percent of the orders coming out of Sales exhibited some type of configuration error.

[11] Because the two systems were based on the same underlying knowledge base, it is difficult to separate out the resources that went into CONFIG versus the resources that were directed toward VERIFIER. From 1981 to 1991, the size of the development group in which the two projects were housed grew from 21 people to 115 people. By 1991, the group's annual operating budget had reached approximately $45 million. By this point, approximately $15–20 million was being spent annually on maintaining VERIFIER and CONFIG, with an estimated 40–50 percent of these funds being spent on CONFIG.

[12] The history and analysis presented here focuses on Jones and Smith because they were identified by many individuals within CompuSys as being the primary decision makers responsible for the escalation of the CONFIG project.

[13] It should be noted that the UDT was not a decision-making body in the sense that it was not vested with the authority to decide whether to continue funding the CONFIG project. However, the UDT meeting minutes did contain critical information concerning the status and ultimate viability of the project. In many cases the decision makers who had the authority to abandon or continue the project were in attendance at these UDT meetings and in all cases, the minutes were widely distributed and available to the key decision makers.

[14] The complete project chronology is available upon request.

[15] The mapping of CONFIG project information was inspired by the work of Newman and Robey who have proposed a similar type of process mapping for analyzing the social character of user-analyst relationships (Newman and Robey, 1992).

[16] The numbers plotted in Figure 6 represent the ratio of the number of times CONFIG was used to generate a quote divided by the number of system quotes generated over the same time period. Information concerning the number of quotes and the number of times CONFIG was used to generate a quote was collected automatically by the company's quoting system. The totals were tallied each month and tracked on a quarterly basis. These figures were supplied to the researcher by CompuSys for the years during which they were available.

References

Allen, T.J. and Marquis, D.G. "Positive and Negative Biasing Sets: The Effects of Prior Experience on Research Performance," *IEEE Transactions on Engineering*

Management (11:4), December 1964, pp. 158–161.

Alter, S. and Ginzberg, M. "Managing Uncertainty in MIS Implementation," *Sloan Management Review* (20:1), Fall 1978, pp. 23–31.

Balachandra, R. and Raelin, J.A. "How to Decide When to Abandon a Project," *Research Management* (23:4), July 1980, pp. 24–29.

Betts, M. "Feds Debate Handling of Failing IS Projects," *Computerworld*, November 2, 1992, p. 103.

Boehm, B.W. *Software Engineering Economics*, Prentice Hall, Englewood Cliffs, NJ, 1981.

Brockner, J. "The Escalation of Commitment to a Failing Course of Action: Toward Theoretical Progress," *Academy of Management Review* (17:1), January 1992, pp. 39–61.

Brockner, J. and Rubin, J.Z. *Entrapment in Escalating Conflicts: A Social Psychological Analysis*, Springer-Verlag, New York, NY, 1985.

Brooks, F.P. *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley, Reading, MA, 1975.

Cohen, J. "A Coefficient of Agreement for Nominal Scales," *Educational and Psychological Measurement* (20:1), Spring 1960, pp. 37–46.

Cringely, R.X. "When Disaster Strikes IS," *Forbes ASAP*, August 29, 1994, pp. 60–64.

Ellis, V. "Audit Says DMV Ignored Warning," *Los Angeles Times*, August 18, 1994, pp. A3, A24.

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

Garland, H. "Throwing Good Money After Bad: The Effect of Sunk Costs on the Decision to Escalate Commitment to an Ongoing Project," *Journal of Applied Psychology* (75:6), December 1990, pp. 728–731.

Garland, H., Sandefur, C.A., and Rogers, A.C. "De-Escalation of Commitment in Oil Exploration: When Sunk Costs and Negative Feedback Coincide," *Journal of Applied Psychology* (75:6), December 1990, pp. 721–727.

Gibbs, W.W. "Software's Chronic Crisis," *Scientific American* (271:3), September 1994, pp. 86–95.

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

Gladden, G.R. "Stop the Life-Cycle, I Want to Get Off," *ACM SIGSOFT Software Engineering Notes* (7:2), April 1982, pp. 35–39.

Hastorf, A.H. and Cantril, H. "They Saw a Game: A Case Study," *Journal of Abnormal and Social Psychology* (49:1), January 1954, pp. 129–134.

Hogarth, R.M. *Judgement and Choice: The Psychology of Decision*, John Wiley & Sons, New York, 1979.

Johnson, J. "Chaos: The Dollar Drain of IT Project Failures," *Application Development Trends* (2:1), January 1995, pp. 41–47

Kahneman, D. and Tversky, A. "The Psychology of Preferences," *Scientific American* (246:1), January 1982, pp. 160–173.

Keider, S.P. "Why Projects Fail," *Datamation* (20:12), December 1974, pp. 53–55.

Keil, M. "Escalation of Commitment in Information Systems Development: A Comparison of Three Theories," *Academy of Management Best Papers Proceedings*, 55th Annual Meeting, Vancouver, British Columbia, August 4–8, 1995a,pp. 348–352.

Keil, M. "Identifying and Preventing Runaway Systems Projects," *American Programmer* (8:3), March 1995b, pp. 16–22.

Keil, M., Beranek, P.M., and Konsynski, B.R. "Usefulness and Ease of Use: Field Study Evidence Regarding Task Considerations," *Decision Support Systems* (13:1), January 1995a, pp. 75–91.

Keil, M., Mixon, R., Saarinen, T., and Tuunainen, V. "Understanding Runaway Information Technology Projects: Results from an International Research Program Based on Escalation Theory," *Journal of Management Information Systems* (11:3), Winter 1995b, pp. 67–87.

Kemerer, C.F. "An Empirical Validation of Software Cost Estimation Models," *Communications of the ACM* (30:5), May 1987, pp. 416–429.

Kindel, S. "The Computer That Ate the Company," *Financial World* (161:7), March 31, 1992, pp. 96–98.

Kull, D. "Anatomy of a 4GL Disaster," *Computer Decisions*, February 11, 1986, pp. 58–65.

Landis, J.R. and Koch, G.G. "The Measurement of Observer Agreement for Categorical Data," *Biometrics* (33:1), March 1977, pp. 159–174.

Lyytinen, K. and Hirschheim, R. "Information Systems Failures—A Survey and Classification of the Empirical Literature," in *Oxford Surveys in Information Technology* (4), P. I. Zorkoczy (ed.), Oxford University Press, Oxford, 1987, pp. 257–309.

Markus, M.L. and Keil, M. "If We Build It, They Will Come: Designing Information Systems That Users Want To Use," *Sloan Management Review* (35:4), Summer 1994, pp. 11–25.

McFarlan, F.W. "Portfolio Approach to Information Systems," *Harvard Business Review* (59:5), September–October 1981, pp. 142–150.

McPartlin, J.P. "The Collapse of Confirm," *Information Week*, October 19, 1992, pp. 12–19.

Mehler, M. "Reining in Runaways," *Information Week*, December 16, 1991, pp. 20–24.

Meredith, J. "Project Monitoring For Early Termination," *Project Management Journal* (19:5), November 1988, pp. 31–38.

Neuman, M. and Robey, D. "A Social Process Model of User-Analyst Relationships," *MIS Quarterly* (16:2), June 1992, pp. 249-266.

Neumann, P.G. and Hoffman, R. "Risks to the Public in Computers and Related Systems: Details of BofA's Costly Computer Foul-up," *Software Engineering Notes* (13:2), April 1988, pp. 6–7.

Newman, M. and Sabherwal, R. "Determinants of Commitment to Information System Development: A Longitudinal Investigation," *MIS Quarterly*, 1996 (forthcoming).

Northcraft, G.B. and Neale, M.A. "Opportunity Costs and the Framing of Resource Allocation Decisions," *Organizational Behavior and Human Decision Processes* (37:3), June 1986, pp. 348–356.

Ropponen, J. and Lyytinen, K. "How Software Risk Management Can Improve System Development: An Explorative Study," working paper, University of Jyvaskyla, Jyvaskyla, Finland, 1993.

Ross, J. and Staw, B.M. "Expo86: An Escalation Prototype," *Administrative Science Quarterly* (31:2), June 1986, pp. 274–297.

Ross, J. and Staw, B.M. "Organizational Escalation and Exit: Lessons From the Shoreham Nuclear Power Plant," *Academy of Management Journal* (36:4), August 1993, pp. 701–732.

Rothfeder, J. "It's Late, Costly, Incompetent—But Try Firing a Computer System," *Business Week*, November 7, 1988, pp. 164–165.

Staw, B.M. and Ross, J. "Commitment to a Policy Decision: A Multi-Theoretical Perspective," *Administrative Science Quarterly* (23:1), March 1978, pp. 40–64.

Staw, B.M. and Ross, J. "Behavior in Escalation Situations: Antecedents, Prototypes, and Solutions," in *Research in Organizational Behavior* (9), B. M. Staw and L. L. Cummings (eds.), JAI Press Inc., Greenwich, CT, 1987a, pp. 39–78.

Staw, B.M. and Ross, J. "Knowing When to Pull the Plug," *Harvard Business Review* (65:2), March–April 1987b, pp. 68–74.

Yin, R.K. *Case Study Research: Design and Methods*, Sage, Beverly Hills, CA, 1984.

About the Author

Mark Keil is an assistant professor of computer information systems in the College of Business Administration at Georgia State University. He received his D.B.A. in management information systems from the Harvard Business School in 1991. His research focuses on the area of software project management, and his articles have appeared in *Sloan Management Review*, *Communications of the ACM*, *Journal of Management Information Systems*, and *Decision Support Systems*.

Appendix A

Excerpts From the Chronology of the CONFIG Project

Date/ Decision Point	Project Information	Coding	Decision/Action
11/81 1:UDT #1	First user design team (UDT) meeting is held. Users are enthusiastic about being involved in the system's development. At the same time, however, they observe many limitations with the prototype system. Observed deficiencies include: no link to company's quoting system, no sizing capability to match customer needs with Compu-Sys hardware/software, and inadequate price information.	Ambiguous	Continue. Users are given remote access to the system and encouraged to provide feedback to the developers as new features are added.
1/82 2:UDT #2	Members of the UDT complain that CONFIG's performance is too slow, that the system lacks sufficient explanation capability, and that the interface is not user-friendly.	Negative	Continue. Developers begin to put emphasis on making the software more robust. Plans are made to have CONFIG operating in Sales offices by 1983.
7/82 3:UDT #3	Members of the UDT raise concerns regarding: timeliness of the product information contained in CONFIG, slow performance of the software, cumbersome user interface, and the need for additional functionality.	Negative	Continue. Developers agree to do a better job of keeping the knowledge base up to date with new products. They also acknowledge that the software is slow and that the performance needs some "fine-tuning." Agreement is reached to load a "production" version of the CONFIG software on an area mini-computer, thus making the software available to approximately 40–60 sales reps outside of the UDT.
1/83 4:UDT #4	Members of the UDT complain that CONFIG is slow, inaccurate, and prone to software crashes. They suggest that the developers put more attention on software testing before they consider releasing it to the field. Users reiterate the need for additional functionality including some type of linkage between CONFIG and price quotation system (PQS).	Negative	Continue. Developers acknowledge many of the issues raised by the users and agree to improve the accuracy and reliability of the software. Other enhancement requests are placed in the category of "ongoing development." Program leader reiterates that "from the [CONFIG] parochial point of view, we will post a non-goal of linking to [PQS] in the first release of [CONFIG]."
7/83 5:UDT #5	Users complain that the development team is not responding to their criticisms of CONFIG. The issue of the timeliness of the product information in CONFIG is again raised. Additional functionality is again requested including a link between CONFIG and PQS.	Negative	Continue. Make additional refinements to the software. Plans made to gradually allow additional users to have access to the CONFIG software.

	When asked about the "quality/readiness" of the software, the users indicate that CONFIG is not yet ready to be released.		
11/83 6:UDT #6	Heated discussion between users and developers. Users accuse developers of not taking their input seriously enough. They continue to complain about the speed and accuracy of the software. Users also continue to raise questions regarding the basic functionality of CONFIG and its ability to communicate with the company's automatic quoting system.	Negative	Continue. A six-month freeze is put into effect on software enhancements, allowing developers to focus on improving the speed and accuracy of the software.
7/84 7:UDT #7	Users express concern over whether U.S. Sales management will support a full-scale implementation of CONFIG. They also complain about the accuracy of the system and observe that there is insufficient hardware in the sales districts for running the CONFIG software. Transcripts of the meeting revealed the following typical comments from users: My perception is that [CONFIG] is totally unused . . . Is it worth continuing with this product? Is this tool helping to solve a problem? The answer is NO. [CONFIG] is not fully developed yet. That's why I feel critical toward it. There was a little bit of support for the product at first but since the implementation process has taken so long people have become discouraged. We are not there yet. There has been improvement but we are not ready yet.	Negative	Continue. Project manager meets with district sales managers to drum up support. Certain members of the UDT agree to meet with the U.S. Sales Management Committee to ask for their support. The committee gives its blessing for a full-scale implementation of CONFIG. Developers acknowledge that the software is slow and inaccurate. Plans are made to equip the districts with dedicated computer hardware for running CONFIG.

Appendix B

Additional Information on Methodology

Data Collection

On-site observations and interviews were conducted during an 11-month period that began in January 1990 and extended through November of the same year. During this period, the developers of CONFIG were in the process of deploying a new version of the software that was designed to be easier to use. This effort marked the last major attempt to adapt and redeploy CONFIG.

Many of the interviews conducted during this period were used to obtain historical details concerning the project. A wealth of historical data on the project was also collected including user design-team meeting minutes as well as other memos and reports documenting various aspects of the project (e.g., system usage reports and cost/benefit analyses).

Following the termination of the project, two additional rounds of interviews were conducted (by telephone) to obtain detailed information regarding why the project was canceled and to understand more about why the project was allowed to continue for as long as it did. These interviews, which were conducted from July through September of 1993 and from April through May of 1994, included discussions with senior managers, sales representatives, sales managers, development project managers, and one of the two key managers who shared responsibility for CONFIG's continuation.

Interview Methodology

The interviews in this research were semi-structured. Before each interview, an interview protocol was constructed and consisted of a list of questions that would be asked of the interviewee. The specific list depended, of course, on the individual's position in the organization and his/her association with the CONFIG project. Most interviews were tape-recorded (except when interviewees requested otherwise) and in all cases copious notes were taken during the interviews. Additional observations were noted immediately after each interview was concluded, and key interviews were then transcribed along with any additional observations.

Most individuals—especially those who had been associated with the CONFIG project during much of its life—were able to offer considerable information regarding why the escalation occurred. In some instances, individuals indicated that they were not well-informed, but were willing to offer their speculation. When an interviewee prefaced his or her remarks with: "I don't know for certain, but my sense is that this is what happened," one could be reasonably certain that the information being offered was of a speculative nature. In these cases, the interviewee was asked to provide the names of individuals who might be better informed on the particular issue at hand. By conducting additional interviews in this way, different respondents' answers could be used for corroboration.

Data Analysis

An analysis of project information that was available to decision makers served as the principle means of establishing that CONFIG was indeed a case of project escalation (i.e., one that involved continued commitment of resources despite negative information concerning the project). As a first step in this analysis, transcripts of interviews and meetings were used to create a detailed history of the project in narrative form. That history was then summarized in the form of a table showing the key project information that was available to decision makers and the resulting decisions or actions that were taken during the course of the project (see Appendix A). After validating this table with several individuals who were familiar with the project's history, the project information available to decision makers was coded as positive, negative, or ambiguous.

To avoid researcher bias, the project information contained in the table (along with the date field and a blank coding field) was shown to two IS doctoral students with project management experience who agreed to serve as independent raters. Neither of the two raters were familiar with the case, and the information they received did not include any reference to decisions or actions taken in response to the project information that was available. For each row in the project information field, the raters were asked to code the information as either purely positive or purely negative. In cases where it was difficult to code the information as purely positive or purely negative, the raters were instructed to use their own judgment and to assign the code that best seemed to fit the information presented and to indicate possible ambiguity by placing a question mark next to the code.

As a measure of interrater agreement, the coefficient Kappa was calculated (Cohen, 1960). A Kappa of 0.72 was obtained, indicating that the strength of agreement between the two raters was "substantial" (Landis and Koch, 1977, p. 163). For the purposes of creating the table that is illustrated in Appendix A, project information was coded as positive when both raters agreed that it was positive, negative when both raters agreed that it was negative, and ambiguous when one rater coded it as positive and the other rater coded it as negative or when either rater indicated that it was hard to classify as either purely positive or purely negative. The result of this analysis showed that the majority of the project information was negative, thereby indicating that the CONFIG project satisfied the definition of project escalation.

The next step of the analysis was to determine the set of factors that seemed to promote escalation and those that seemed to be associated with the eventual termination of the project. In the case of CONFIG, a

model of escalation based on the categorization of factors discussed by Staw and Ross (1987a) was used as the basis for identifying and organizing the factors that seemed to promote escalation. The first step in this analysis involved comparing and contrasting the CONFIG case against the array of factors that have been discussed in the escalation literature and noting which of these factors seemed to be present in the case. The next step involved the identification of additional factors that were present in the CONFIG case but had not been widely discussed in the escalation literature.

The entire analysis process was highly iterative. Before a factor was identified as a possible cause of the escalation, a considerable amount of cross-checking of interview transcripts was performed to verify that there were at least two or more sources of evidence in support of that factor. This approach was a variation on the pattern matching technique for case analysis advocated by Yin (1984). A similar approach was used to identify the factors that seemed to be associated with the eventual termination of the project. After identifying a set of factors associated with both the escalation and eventual termination of the project, the respective factors were reviewed with several contacts at the case site as a means of validation.

Limitations of the Research Approach Used Here

While the general approach described above has some clear strengths, there are also some limitations that should be noted. The most significant limitations concern generalizability and the inability of knowing what went on inside the heads of key decision makers such as Jones and Smith.

Clearly, there is a limitation concerning the generalizability of a single case study. As a general rule, it is preferable to use a multiple case-study design in which theoretical and/or literal replication is possible. Given the sensitivity that surrounds most failing projects, however, the researcher may seldom have the opportunity to structure the type of multiple case-study design that would be desirable. Instead, one must sometimes follow a more opportunistic approach even if that means settling for a single case study.

The second major limitation of this study is the inability of knowing what went on inside the heads of key decision makers such as Jones and Smith. Without knowing this, it is impossible to determine the relative strength of the various factors that may have contributed to the escalation. A more robust analysis would require: (1) better access to those most responsible for the escalation, and (2) the ability to read their innermost thoughts and motivations concerning the project. In this study, the researcher had only limited access to key decision makers such as Jones and Smith, the two individuals who were among CONFIG's strongest advocates and therefore most able to provide insight concerning its escalation.

In order to overcome this limitation, this research relies heavily on the observations and insights of individuals who were closely associated with the project and in a position to comment on the motivations and behaviors of Jones and Smith. The triangulation used in the study, which included extensive discussions with the managers to whom Jones and Smith reported, allowed the researcher to develop a rich understanding of the factors that promoted project escalation.

In summary, while there are some limitations associated with the approach used here, the results of this research still provide some useful, if tentative, findings that should be of interest to both researchers and practitioners.