Exploring Re-escalation Cycles in Systems Implementation: Extending Project Escalation and De-escalation Theories

Pisit Chanvarasuth [1)], Mark R. Nelson [2)], Thiagarajan Ravichandran [3)]

[1)] Ph.D. Candidate, Rensselaer Polytechnic Institute (chanvp@rpi.edu)
[2)] Assistant Professor, Rensselaer Polytechnic Institute (nelsom@rpi.edu)
[3)] Assistant Professor, Rensselaer Polytechnic Institute (ravit@rpi.edu)

Abstract

Traditionally, escalation theory focuses on the continued commitment to a previously chosen course of action in spite of negative feedback concerning the viability of that course of action [12], whereas de-escalation theory focuses on the factors or processes by which commitment to a previous course of action is reduced [15]. While the strength of escalation behavior varies from one culture to another [16], there is sufficient evidence to suggest that these phenomena exist internationally and in both government and business organizations [13, 16, 21]. Several lenses have been used to study these phenomena, including but not limited to, escalation of conflict [26], entrapment [2], escalation of commitment [1, 12, 22], abandonment theory [7, 8], and project redirection [15, 18].

Despite the existing literature, there are still gaps in our understanding of these phenomena. Much of the existing literature examines escalation and de-escalation independent of one another. These studies also focused on projects of a "one-time" nature that once terminated, did not begin again. This paper provides a review and interpretation of the escalation and de-escalation literature with an eye towards developing a model that spans the escalation to re-escalation cycle in large-scale systems implementation. As part of this process, two examples illustrating the re-escalation cycle are discussed.

Key Words: escalation theory, de-escalation theory, IT project failure, systems implementation

1. Introduction

Each year organizations around the world, both public and private, spend a combined total of billions to trillions of dollars implementing systems. Consistent with the increasing investment patterns oriented towards implementing large-scale systems, such as ERP and global supply-chain integration systems, IS researchers are beginning to recognize the need for more research into the processes that lead to the success or failure of these projects. Unfortunately, available evidence indicates that large-scale IT projects have failure rates that exceed 50 to 70 percent [5, 20]. In recent years, there has been a growing interest in systems implementation failure, especially processual dimensions of systems implementation, such as the processes of escalation and de-escalation. Many of these large-scale projects escalated beyond original budget and schedule constraints during implementation before eventually de-escalating to the point of termination. However, the strategic importance or technological imperatives underlying these projects often require that they not be permanently terminated, leading them to be restarted and often to once again escalate out of control, as shown in Figure 1.

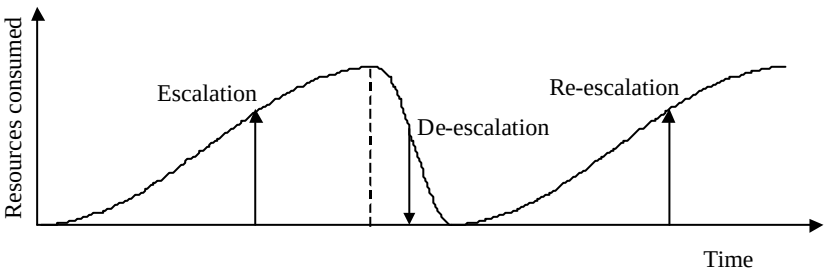

Figure 1 Projects are possibly re-escalated after terminating the previous cycle

While several studies examine large-scale projects that have escalated (e.g., [13]) or de-escalated (e.g., [10, 18]), previous studies focused primarily on either the escalation or de-escalation process alone. These studies also tended to look at projects with a "one-time" nature, rather than projects where a new implementation cycle must begin if the previous cycle failed. For these reasons, we have less understanding of how these two processes fit together to explain the overall process, or what happens in situations where de-escalation does not end in complete termination due to contextual factors or imperatives that make complete and permanent project termination unfeasible. Thus, gaining a

better understanding of the re-escalation cycle will help organizations learn from their experiences and reduce the chance of project escalation in subsequent project iterations.

This paper is organized as follows: the next sections present a very brief review of the IS literature on large-scale systems implementation, including a definition of large-scale. The next sections discuss alternative views of project escalation and project de-escalation in the literature as they relate to systems implementation. The following sections extend these concepts to illustrate the phenomenon of project re-escalation, which sometimes occurs in systems implementation through both case studies and discussion. The final section provides a summary and discussion of future research directions on re-escalation cycles in systems implementation.

2. Large-Scale Systems Implementation

Large-scale IT projects are notoriously difficult to control, with reported failure rates that exceed 50 to 70 percent [5, 20]. A common characteristic of many of these projects is that they tend to escalate beyond original budgets and completion schedules, de-escalate or are terminated, and then restart or continue, only to escalate again. Our understanding of this phenomenon is somewhat limited, as the research on large-scale systems is insufficiently broad, both in theoretical understanding and methodological approaches [5] and includes gaps in our understanding of processual and contextual dimensions of the implementation process [17].

When discussing large-scale systems implementation, there are a variety of definitions for "large-scale." This variety stems from the difficulty in establishing a single variable or limited set of variables that define "largeness" in general [3, 28]. Zmud [28, p. 15] defined a large-scale system "to be one that requires more than one management level to coordinate the development effort, and more than a six month development period." Sundgren [25] suggested that largeness could be measured by the amount of information that is available in the system and the amount of information processing that is going on in the system. The Standish Group [20] suggests that any system development project costing over $10 million is large-scale. Combining these definitions, we could conclude that large-scale systems are defined by the degree of *complexity* involved as defined by the size of the information sets and machine programs, and the amount of information management resources required to implement the system.

3. Escalation Theory

While there are many different modes of IT failure, one pattern of failure that has been observed is the systems implementation project that seems to "take on a life of it own," continuing to consume valuable resources without ever reaching its objective [13]. Eventually, these projects are abandoned or redirected, but the cost of having funded them can represent an enormous waste of organizational resources. Many IT projects that seem to take on a life of their own represent what can be explained as escalation.

In psychological analysis, escalation is the continued commitment to a previously chosen course of action in spite of negative feedback concerning the viability of that course of action or persistence of a venture beyond an economically defensible point [1, 6, 14, 22]. Early research in this area focused on behaviors where negative feedback following the choice of some action stimulated concern for the justification of that choice in the hope that future positive outcomes might vindicate the original choice (self-justification theory) [10]. There have been three major research streams building in this field: (1) escalation of conflict [26]; (2) entrapment, where individuals are likely to expand resources toward a receding or elusive goal [2]; and finally, (3) escalation of commitment [1, 21, 22]. Research on individual decision making has concluded that people tend to escalate commitment in response to previous investment. Staw [21] demonstrated that subjects who were both committed to an investment and failed subsequently invested more money than subjects who were not committed or who did not fail. Staw and Ross [24] further proposed a useful taxonomy that groups possible escalation factors into four categories: project factors, psychological factors, social factors, and organizational factors.

Other research in IS implementation suggests that many failed large-scale IT projects exhibit characteristics indicative of escalating commitment to a failing course of action and that theory explaining this phenomenon might usefully be applied in the context of IT failure [12]. Like in psychology, escalation of system implementation projects represents what can be described as "continued commitment to a failing course of action" [13]. A key motivator of such escalation may be the concealment, distortion, or avoidance of negative information about the project [12]. Several approaches have been used to study this phenomenon, as illustrated in Table 1.

Table 1: Approaches to studying escalation theory

Construct	Variables	Findings	Studies
Approach Avoidance	Ambiguity	Ambiguity increases conflict, leading to more pressure to persist.	[2]
	Reward	People tend to persist longer when there are only a small decrement of resources over time	[2]
	Proximity	The motivation to achieve a goal increases as an individual gets closer to that goal	[4, 12]
Self-Justification	Dollar allocated, Personal responsibility, Involvement	People tend to act consistently with past action but will eventually de-escalate; Likewise, people who feel involved in a failing course of action escalate their commitment	[2, 12, 21, 22, 23, 24]
Social Determinants	Commitment of resources, large audience-high social anxiety condition	Decision makers may persist in a course of action not only because they do not want to admit to themselves that they made a mistake, but also because they may be hesitant to expose their errors to others	[23, 24]
Prospective Feedback	Efficacy of resources	More resources are allocated when the report was positive but less after negative feedback; Decision makers faced with many choices will prefer the relative uncertainty of the future loss to the relative certainty of the present loss and thus elect to continue a failing project	[23, 27]
	Sunk cost	Sunk cost contributes positively to escalation. The decision makers' use of sunk cost information correlates with escalation behavior	[11]
Goal Congruency	Degree to which the primary decision makers (agent) and senior management (principle) have the same goal	The agent may suffer negative career consequences for reporting bad news, which leads to project escalation	[12]
Information Asymmetry	Degree to which the principal has less information than agent	Whenever the primary decision maker conceals or distorts negative information, or put a positive spin on any negative information, escalation can be promoted	[12]
The Role of Experience	Experience	Experience is associated with lower escalation tendency. Problem- and context-specific expertise can have a powerful effect on decision-making behavior	[10, 26]

4. De-escalation Theory

In contrast to escalation theory, de-escalation examines the factors or processes by which commitment to a previous course of action is reduced. While prior research has shown that managers can easily become locked into a cycle of escalating commitment to a failing course of action [18, 26], there has been comparatively little research on de-escalation, or the process of breaking such a cycle. For IT project failure, de-escalation is often equated with abandonment theory where commitment to failing IT projects is reduced to the point of project termination [7, 8]. However, others have defined de-escalation more broadly to include not only project abandonment, but project redirection as well [18]. Recent work in this area has begun to suggest processes for de-escalating runaway projects; however, it also notes that much additional work is needed in this area [14, 15].

Table 2: Conditions that promote de-escalation

Condition	Findings	Relevant studies
Administrative Turnover	Administrative turnover creates change by effecting changes in the decision making, at the same time change is brought on by the mere symbolic value of it.	[13, 24]
Bifurcated Decision Procedures	Bifurcation of the decision making process into initial and subsequent phases, is a technique that can reduce commitment.	[24]
Unambiguous Negative Feedback	To guard against positive biasing as information filters up through hierarchy, multiple channels need to be developed. It is all the more important since presentation of unambiguous data is difficult in organizational settings. Unambiguous manifestation of the problem is necessary for de-escalation to take place.	[11, 12]
Setting Minimum Target Levels	When minimum target levels are set as criteria for success, troubled projects tend to de-escalate.	[12, 15]
Support for Failure	People tend to defend the losing course of action even more under the threat of demotion or job loss. De-escalation can be encouraged by not imposing severe punishment for failure.	[12, 15, 24]
Deinstitutionalization	Removing the project from the core of the firm can facilitate de-escalation.	[12, 24]
Regular Evaluation of Projects	Routine reviews and evaluations tend to encourage de-escalation.	[12, 15]

Of the existing de-escalation studies, multiple approaches have emerged and Table 2 summarizes the known conditions that trigger de-escalation. Using a variance approach, Keil and Robey [15] identified twelve factors drawn from prior empirical research. These include: change in top management support, external shocks to the organization, change in project champion, organizational tolerance for failure, the presence of publicly stated resource limits, consideration of alternative uses of funds supporting a project, awareness of problems facing the project, visibility of project costs, clarity of criteria for success and failure, organizational practices for evaluating decision makers, regular evaluation of projects, and separation of responsibility for approving and evaluating projects. However, each of these twelve factors offers only a partial explanation for the de-escalation of commitment to troubled project. In a process-oriented approach, Montealegre and Keil [18] suggested that de-escalation occurs as a four-phase process: (1) problem recognition, (2) re-examination of prior course of action, (3) search for alternative course of action, and (4) implementing an exit strategy. Their findings suggest that an escalated project is not simply abandoned or immediately de-escalated upon receipt of unambiguously negative information. Instead, a project goes through several distinct phases along the road to de-escalation and that there are key triggering activities associated with each of these phases.

5. Two Examples of Re-escalation

In this section, two projects are presented to illustrate the re-escalation cycle that sometimes occurs in systems implementation. The two examples are: the Taurus project at the London Stock Exchange, which was in development from 1981 through 1993; and the Tax Systems Modernization (TSM) project at the Internal Revenue Service (IRS) which began in the 1980s and is still going today. These brief project summaries were developed through review of publicly available documents and reviews. The Taurus summary was developed from several sources, consisting of [6, 9, 14]. The TSM summary was produced from a review of over 700 government documents on the project made available through the United States General Accounting Office [19]. Both of these projects resulted in final expenditures in the millions (or billions) of dollars beyond original budgets and timelines years beyond original completion schedules. Both of these projects involve large organizations employing or interacting with thousands of employees across a variety of physical locations. Whether one considers time, costs, application size, or complexity, these projects represent very large-scale systems implementations. Interestingly, both of these projects experience at least one re-escalation cycle, which we believe to be common in many very large-scale systems projects.

Taurus. Taurus (Transfer and Automated Registration of Uncertificated Stock) was a major IT project at the International Stock Exchange (ISE) for the dematerialization of shares. The introduction in 1979 of Talisman, a system for settling share deals that was entirely computer based led to the proposal in 1981 to extend automation to the market as a whole and to end the use of paper share certificates. The Taurus project proposed that information on shares and shareholders would be located in a single large database that would be administered by the Stock Exchange, rather than across professional registrars or individual companies. Under this proposal the procedures surrounding the buying and selling of shares would be streamlined and simple account transfers would achieve changes in ownership. Despite opposition, the Stock Exchange continued to work on the project until late 1988 when it was concluded from independent reviews that the project in its current form should be canceled. In addition to the opposition from vested

interests, the £60 million already spent was considered to be too high and the technical complexity too great. Eventually, in September 1988, after the review of Taurus was undertaken, the Taurus project was abandoned due to projected cost and widespread opposition from Share Registrars.

However, the Taurus project did not end with its first abandonment. Soon after, the Bank of England became involved and was instrumental in the creation of an industry-wide committee. By Spring 1989 the committee recommended that the best way forward was to expand the existing Talisman system at a projected cost of nearly £50 million. After extensive changes to the design, the Taurus was re-launched in March 1990. After a few delays and project postponements, a new target completion date was set with a projected cost of £15 million added to the original £50 million. By January 1993, a full review of the project was ordered, revealing a series of problems that would result in postponing the program testing for at least 15 months. As a result of the review, it was estimated that Taurus would take another three years to build and that costs could be double. In March 1993 the Board of the London Stock Exchange decided to terminate the Taurus project. At the point of abandonment at the end of the second escalation and de-escalation cycle, the London Stock Exchange had spent over £80 million for Taurus project and individual securities firms in London had spent an estimated £400 million to prepare their own systems for Taurus. The project had escalated out of control for a second time, and again resulted in project abandonment.

TSM. The TSM project at the IRS involves the integration of over 50 applications on several different hardware and software platforms, some of which date back to the 1950s and are still in use as of 2001. The systems involved require both a large amount of information be available and an intensive amount of information processing take place. Audits by the United States General Accounting Office (GAO) also uncovered serious weaknesses within the agency's financial systems, including cumulative discrepancies in excess of $30 billion. After three false starts in the early 1980s, the TSM project began in earnest in 1986. In 1997, a decade and over $3.4 billion later, the TSM project was terminated by the new agency commissioner, Charles Rossotti. At that time, the TSM project was under close scrutiny for being over-budget, behind schedule, and with many or most objectives unfulfilled. Documents from inside and outside the agency indicate that over time a variety of technological and managerial factors impeded TSM's progress. However, complete abandonment was not an option for the agency, where an antiquated and aging IT infrastructure had to keep pace with a growing volume of tax returns and address serious data accuracy and security weaknesses. Thus, within a very short span of time, the TSM project began again with an initial estimated cost of $5 billion to $7 billion and a 10 to 15 year project schedule. Nearly four years into that project there are already signs provided through GAO audits and testimony to Congress that the project may again be escalating.

6. Discussion

Based on data from these two case studies, we developed the concept of a re-escalation cycle in systems implementation. Even though the Taurus project was temporarily abandoned in late 1988, when revived the project escalated a second time resulting in a second termination of the project in 1993 after finding expenditures and schedules greatly exceeding original expectations. However, this evidence contradicts Staw and Fox [23] who argued that there is little evidence that escalation persists over repeated investments and that once engaged de-escalation processes should prevent further escalation as investment continues. We propose that such re-escalation is probably common in very large-scale system implementations, where outright failure is not a lasting option. Take, for example, the IRS TSM project. The agency has a critical imperative to modernize, driven both by an increasing volume of tax returns and data inconsistencies, which the current systems are unable to support or resolve. Such large projects are not undertaken lightly by organizations. They normally incorporate significant business or organizational imperatives which may make the projects an issue of survival or operational viability, such as the need to increase processing capability or replacing aging legacy infrastructures. In these situations, de-escalation or abandonment of projects may only be temporary because the underlying business imperatives that originated the need for such large projects likely persist independently of the success or failure of particular project iterations.

The fact that large-scale projects such as Taurus or TSM may re-escalate raises several interesting questions for researchers. First, when previously escalated projects de-escalate and continue or are abandoned and restarted, how often do they result in re-escalation? Are there large-scale projects that are successful at preventing re-escalation, and if so, how is this accomplished? Some theories [10, 26] suggest that experience or other factors may help reduce the tendency for such projects to re-escalate. By looking at situations of re-escalation, we may be able to develop new theories about the persistence and importance of various de-escalation factors and processes. In short, we need more evidence to confirm the existence of re-escalation cycles and the processes that promote or prevent such cycles from occurring.

Second, if re-escalation does occur, are the escalation factors the same as in previous iterations or do new phenomena or factors emerge that promote escalation in second or subsequent iterations? While we are aware of different factors that influence escalation or re-escalation, we do not know if the roles or importance of those factors differ after the first escalation/de-escalation cycle. For example, the escalation literature on social determinants [23, 24], prospective feedback [11, 23, 27] and self-justification [2, 12, 21, 22, 23, 24] all provide theoretical constructs which might suggest

that the risk of escalation is potentially higher in new project iterations than in an original escalation cycle because both visibility of failure and the amount of resources spent are higher. By understanding the factors and processes involved with re-escalation, we may come to better understand escalation phenomena in general.

Finally, we know that large-scale projects experience high failure rates [5, 20]. This suggests that such projects may be a good context for studying the relationship of escalation and de-escalation factors and processes to project failure. Furthermore, as noted before, such large-scale projects are probably not undertaken lightly due to the anticipated resource requirements. Thus, these projects are most likely closely tied to requirements that represent strategic, operational or environmental imperatives for an organization. The study of re-escalation cycles in these projects may introduce new theories of escalation, de-escalation and project management, particularly in relation to strategic alignment.

7. Summary

In this paper, we presented the concept of project re-escalation cycles in systems implementation. The evidence from our examples illustrates how previously de-escalated projects may re-escalate. Gaining a better understanding of the re-escalation cycle will help organizations learn from their experiences and reduce the chance of projects re-escalating in subsequent iterations. This is particularly important in large-scale projects where the costs of escalation may run in the millions to billions of dollars. The concept of re-escalation cycles introduces a number of interesting research questions and we hope that the ideas introduced in this paper will stimulate interest and future work by IS researchers.

References

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

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

[3] Brussard, B., "Large Scale Information Systems: A Comparative Analysis," in Informatization Developments and the Public Sector, European Public Administration and Informatization Series, Vol. 2, Ed. P. Frissen, Amsterdam, The Netherlands: IOS Press, pp. 171-183, 1992.

[4] Conlon, D.E. and Garland, H. "The Role of Project Completion Information in Resource Allocation Decisions," *Academy of Management Journal*, (36:2), pp. 402-413, 1993.

[5] CSTB, *Making IT Better: Expanding Information Technology Research to Meet Society's Needs*," Committee on Information Technology Research in a Competitive World, Computer Science and Telecommunication Board, National Research Council, National Academy Press, 2000.

[6] Drummond, H., Escalation in Decision-Making: The Tragedy of Taurus, New York, NY: Oxford University Press, 1996.

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

[8] Ewusi-Mensah, K. and Przasnyski, Z. H. "Learning from abandoned information systems development projects," *Journal of Information Technology*, (10), pp. 3-14, 1995.

[9] Flowers, S. Software Failure: Management Failure, New York, NY: Johnson Wiley & Sons, 1996.

[10] Garland, H. "Throwing Good Money After Bad: The Effect of Suck Costs on the Decision to Escalate Commitment to an Ongoing Project," *Journal of Applied Social Psychology*, (75), pp. 728-731, 1990.

[11] Garland, H. and Conlon, D.E. "Too Close to Quit: The Role of Project Completion in Maintaining Commitment," *Journal of Applied Social Psychology*, (28:22), pp. 2025-2048, 1998.

[12] Keil, M., Mann, J. and Rai, A. "Why Software Projects Escalate: an Empirical Analysis and Test of Four Theoretical Models," *MIS Quarterly*, (24:4), pp. 631-664, 2000.

[13] 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), pp. 65-85, Winter 1994-1995.

[14] Keil, M. and Montealegre, R. "Cutting Your Losses: Extricating Your Organization When a Big Project Goes Awry," *Sloan Management Review*, pp. 55-68, Spring 2000.

[15] Keil, M. and Robey, D. "Turning Around Troubled Software Projects: An Exploratory Study of the De-escalation of Commitment to Failing Courses of Action," *Journal of Management Information Systems*, (15:4), pp. 63-87, Spring 1999.

[16] Keil, M., Tan, B.C.Y, Wei, K., Saarinen, T., Tuunainen, V., and Wassenaar, A. "A Cross-Cultural Study on Escalation of Commitment Behavior in Software Projects," *MIS Quarterly*, (24:2), pp. 299-325, 2000.

[17] Montealegre, R. "What Can We Learn from the Implementation of the Automated Baggage-Handling System at the Denver International Airport?" 1996 Americas Conference, Association for Information Systems, Phoenix, AZ, August 16-18, 1996.

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

[19] Nelson, M. and Ravichandran, T. "Understanding the Causes of IT Failures in Government Agencies," 2001 Americas Conference, Association for Information Systems, Boston MA, August 2001.

[20] Standish Group, *CHAOS: A Recipe for Success*, res. rep., see www.standishgroup.com, 1994.

[21] Staw, B. M. "Knee-deep in the big muddy: a study of escalating commitment to a chosen course of action," *Organizational Behavior and Human Performance*, (16), pp. 27-44, 1976.

[22] Staw, B. M. "The Escalation of Commitment to a Course of Action," *Academy of Management Review*, (6), pp. 577-587, 1981.

[23] Staw, B. M. and Fox, F.V. "Escalation: The Determinants of Commitment to a Previously Chosen Course of Action," *Human Relations*, (30), pp. 431-450, 1977.

[24] Staw, B.M. and Ross, J. "Behavior in Escalation Situations: Antecedents, Prototypes, and Solutions," Research in Organization Behavior, (9), pp. 39-78, 1987.

[25] Sundgren, B., "The Elements of Largeness," in Trends in Information Systems, Eds. B. Langefors, A. Verrijn-Stuart, and G. Bracchi, New York: North-Holland Publishing, pp. 23-33, 1986.

[26] Teger, A. Too Much Invested to Quit, New York, NY: Pergamon Press, 1980.

[27] Whyte, G. "Escalating Commitment to a Course of Action: A Reinterpretation," *American Management Review*, (11:2), pp. 311-321, 1986.

[28] Zmud, R., "Management of Large Software Development Efforts," *MIS Quarterly*, (June), pp. 45-55, 1980.