

Escalation in IT Projects: Can We Afford to Quit or do We Have to Continue?

Urban Nulden

Department of Informatics, Goteborg University, Sweden

email: nulden@adb.gu.se

Abstract: Many information technology (IT) projects fail. These projects are not within budget, not on time or do not deliver what was promised. Failures in IT projects are more common than failures in any other aspect of modern business. Whereas a large number of projects are obvious failures, some are not. A model of escalation—increasing commitment to a failing course of action—is used to analyze and explain one type of project failure, namely, those projects that seem to take on a life of their own, wasting scarce resources and in many cases never reaching the objectives they were set to fulfill. This paper argues that escalation as a cause of IT project failure has received too little attention in both practice and education. A small case is used to illustrate escalation in an IT project.

Nuldén, U. (1996). Escalation in IT Projects: Can We Afford to Quit or do We Have to Continue. Information Systems Conference of New Zealand, Palmerston North, New Zealand, IEEE Computer Society Press, p.136-142.

1. Introduction

How many times have you been in the situation where you spend two hours on a bad movie and feel that with this time invested it would be a waste of time to leave the TV-set before the movie is finished. You say: I have “too much invested to quit” [1]. This paper discusses this behavior in failing information technology (IT) projects.

Traditional wisdom tells us that project failure depends on lack of control and poor project management. That is probably true, but what is behind the superficial explanation of poor project management? This phrase has become a dumping ground for a plethora of different possible explanations to failing IT projects, such as inadequacies of identifying requirements, ignorance, chronic tendency to underestimate cost and scope of a project, unsuccessful management of the risk associated with IT, user dissatisfaction etc. [2, 3].

While there are many types of failures, some IT projects seem to take on a life of their own, “they become like Moses, condemned to wander till the end of their days without seeing the promised land” [4]. We call these projects “runaways” as they continue to absorb valuable resources without ever reaching their objectives [5, 6]. Eventually these projects are terminated. There is, however, a tendency in IT projects to let a “runaway” carry on way too long before action is taken. Organizations that continue to put resources in a runaway project err since the additional money spent does not give the business effect the project was intended to accomplish, the organization is wasting valuable resources, and the resources wasted here could probably be better invested somewhere else [2]. Moreover, abandonment of an unsuccessful IT project can be a positive outcome for the organization. It provides the organization with a chance to learn from its mistakes in projects and thus minimizes the risk of making similar mistakes in the future [7].

This paper argues that escalation as an explanation of project failures have had too little attention. The remainder of the paper is structured as follows: The next section discusses the problem of determining when a project is a failure. Section three and four discuss commitment as a factor determining failure or success and a model of escalation is discussed. In section five, a case study is used to illustrate escalation in a systems development project, ADMIN. Finally, there are recommendations for practitioners and organizations involved in IT projects, and some implications for education of IT professionals.

2. When Projects Fail

There are no consistent notions of what precisely is meant by failure and what is meant by success of an IT project. Instead it is open to considerable interpretation and likely to be

viewed differently by different persons at distinct times. For instance, the users can be very satisfied with a new a computer system, but if the cost came to be twice or three times the budgeted, management would probably have another opinion. Some argue that we need to redefine success since so many projects fail in some major way [8]. The concept and understanding of IT project failure have changed over the years from technical issues to management, organizational and social issues. However, research seems to agree that the level of failure is far too high and that this is a continual problem [9]. The IT literature contains a wealth of ideas, methods, and suggestions for how to reduce the risk of failure by improving the state of art in development and project management, but the literature also explicitly ignores any consideration of failure reasons [10].

Consider a scenario where a consulting firm in 1988 was to develop a new information system for a Hilton Hotels, Marriott, and Budget Rent-A-Car consortium [11]. The system, CONFIRM, was to be a state of the art comprehensive travel industry reservation system. Three and a half years after the project start and when a total of \$125 million had been invested, the project was canceled. One senior IS manager wrote:

“Some people who have been part of CONFIRM management did not disclose the true status of the project in a timely manner. This has created more difficult problems—of both business ethics and finance—than would have existed if those people had come forward with accurate information. Honesty is an imperative in our business—it is an ethical and technical imperative.”

Clients and senior management were misled into continuing to invest in a project plagued with problems in database, decision support, and integration technologies. People in the project knew this long before the project was canceled but no one came forward with this information, i.e., no one “blew the whistle” [12]. A possible explanation for the failure of CONFIRM is that people in the project, e.g., the project manager, were too committed and confident that problems would be solved. But, on the other hand, IT projects will surely fail if people are not enough committed to their project. Although, there is a very thin line between an optimistic attitude and overcommitment.

3. Commitment to a Project

Commitment has been studied from so many different theoretical perspectives that the concept should be abandoned in favor of a set of concepts [13]. However, in this paper, commitment is itself not necessarily good or bad, but the level of commitment of various individuals in a project is believed to greatly influence the eventual success of the project. Commitment is defined as the binding of an individual to behavioral acts [14], or in other

words, the state of mind that holds individuals in a line of behavior [15]. Moreover, commitment can also be described as an active counterforce to change [16].

For the purpose of understanding commitment in organizations it is important to emphasize that when commitment induces a person to complete a difficult or unpleasant task that benefits him and others, then commitment is a good thing. When, however, commitment leads to fixation on a policy or behavior of diminishing benefit and rising cost, the situation is obviously less clear. We have overcommitment, or in other words, we have an escalation situation.

4. Escalation Situations

An escalation situation is when decision makers become overcommitted to previous decisions, and invest more resources in a failing project. This situation occurs when decision makers have continued commitment to a specific course of action despite information suggesting that the course of action is failing [17, 18]. Brockner [19] elaborates this further by arguing that an escalation situation is continued commitment in the face of negative information about prior resource allocations coupled with “uncertainty surrounding the likelihood of goal attainment.”

Decision makers become locked in an escalation situation through what Staw [16, 17] calls a “syndrome of decision errors.” This is criticized by Bowen [20] who argues that commitment to further investment occurs because of the equivocality of the situation and not because of a commitment to a failed decision. Bowen continues that “technically” one cannot err in an ill-structured decision situation.

There is controversy concerning the explanation of escalation. Brockner [19] argues that many, but not all, of the explanations fall into two broad categories explained by expectancy theory and self justification theory. Expectancy theory posits that individuals have a tendency to believe that allocating additional resources will eventually lead to goal attainment. Self justification theory explains the individual’s desire to demonstrate rationality to himself because people do not like that their past decisions were insufficient. However, all escalation situations have some common factors that can be isolated [18]: (1) all situations entail some loss or cost—not necessarily monetary—that has resulted from an original course of action, (2) the predicaments involve some continuity over time—they are not one-shot affairs, but dilemmas involving ongoing courses of action, and (3) they comprise situations where a simple withdrawal is not an obvious solution. Moreover, (4) the decision maker must have a real choice in deciding whether to persist or withdraw [19], and (5) there must be unambiguous feedback from previous decisions made [20].

Simultaneous with the research on escalation, conducted primarily by organizational behavior researchers, social psychologists have studied the same phenomenon using the term “entrapment” [21]. Entrapment situations are those in which decision makers continue investing their resources in a costly or losing course of action in order to justify the appropriateness of already sunken costs. Although entrapment refers to a decision making process, it is often the outcome of that process that is particularly noteworthy. That is, in the process of justifying already committed resources, the individual can be drawn into an extremely costly or even irrational course of action [22]. In contrast to escalation situations, subjects in entrapment situations typically incur small continuous losses as they seek or wait to achieve a goal [20].

To avoid losing the essence of escalation as a phenomenon, attention should be shifted from identifying the large number of isolated antecedents of escalation situations to examining the influence of more general classes of determinants in a range of situations.

4.1 A Model of Escalation

Staw and Ross [23] propose an escalation model which gives us a promising theoretical base for analyzing and to some extent explaining what “poor project management” is. They group the antecedents in four classes of determinants of escalation situations: project determinants, psychological determinants, social determinants, and organizational determinants.

Project determinants are the objective attributes of a project, the project’s benefits and costs [19]. A project is likely to be continued with high commitment if it is perceived as a long-term investment, expected to have a large payoff, and a long-term payoff structure [24]. High commitment to a project is also more likely to occur when closing costs are high and salvage value is low [23].

Psychological determinants cause individuals to take an optimistic view [19]. Psychological determinants explain people’s unwillingness to admit that an earlier decision was wrong [18]. Self justification theory and prospect theory—individuals exhibit risk averse or risk seeking behavior depending on how a problem or decision situation is framed—are psychological theories explaining escalation [25, 26]. Another psychological determinant is the human irrational economic behavior to “throw good money after bad” in an attempt to turn around a failing situation, the so called “sunk cost” effect [27, 28]. People who have initiated a project and are held responsible for success or failure are more likely to continue that project [29]. Moreover, the more public a decision is made, the less likely it is for the decision maker to change his original decision.

Social determinants originate from the group of which the individual is a member and hold the individual to a course of action regardless of the individual’s own beliefs. Examples are

face saving and external justification [18]. Social comparison theory posits that people evaluate their attitudes and behavior in relation to others and are likely to regard the behavior of others as a model for their own behavior [30]. This evaluative process is most apt to occur when decision makers are uncertain about the appropriateness of their own attitudes or behavior. Social determinants also involve a group's relation to another group. A successful effort by a group may influence another group to attempt the same approach. Moreover, the behavior of project members are vitally affected by their relative power position [31].

Finally, there are *organizational determinants* in the structural and political environment of a project, such as top management support, administrative inertia, power and interorganizational interaction. According to Keil [2], projects are more prone to escalate when there is strong political support and when projects become institutionalized. Institutionalization occurs when a project is tied integrally to the values and purposes of the organization, and when actions are taken for granted because they are so deeply imbedded in the subculture or norms of the organization. Long-standing programs and lines of business are not even considered for discontinuation because they are so identified with the organization.

5. A Case Study

Investigation of troublesome or failing IT projects presents special problems due to the sensitive nature of the subject matter. Most people tend to hide, forget or put the failure behind as fast as possible and move on to the next project [8]. Also, there seems to be a code of silence in the software development community which prohibits discussions of failures [7]. The short case described in the next section serves as an example of a project that exhibits characteristics consistent with the escalation framework presented in the previous section. It should be stressed that facts are omitted to keep the description of the project brief. The project is still in progress and no complete evaluation or audit has yet been done.

5.1 The ADMIN Project

In 1994 a subdivision of a mid-sized company began developing a computer based system called ADMIN. The aim of the system is to improve a relatively simple but very important administrative task. The success of the subdivision, and in the long run the whole organization, is to some extent dependent on this administrative task and that it operates smoothly.

Before the project started, during winter and spring the same year, a system developer from the central systems support department got the request from a manager in the subdivision: Could he put together a small computer application that would support the administrative

task. The developer agreed and started a “quick-n-dirty” attempt, but soon realized that this computer system would be a central part of the subdivision and therefore suggested a more formal and structured approach for ADMIN. The actual project started in May 1994. The initial project group consisted of three persons. Two persons from the department carrying out the task, hereafter referred to as users, and a systems analyst from the local systems support department. Officially one of the users was project manager, but in reality the systems analyst managed the project. This was his first project as project manager. The users had no prior experience of working in a software development project.

The analysis phase of the project was completed in June, followed by the systems requirements and specification phase which was completed in February 1995. The work in these phases encountered no major problems. A client/server solution was suggested since ADMIN was to be used at two different sites. An external consultancy firm, hereafter referred to as the firm, were engaged for the programming part. This firm had been involved in successful projects as well as some less successful projects earlier, but had a reputation to be competent. During the discussion of technical issues of ADMIN one important demand was raised. The tools to be used in the project should be approved by the central systems department. The department had experience of ‘Access’ and ‘Visual Basic’ and found these appropriate for ADMIN. Later this showed to be a bad decision.

The development phase was officially managed by the systems analyst and originally scheduled to be completed in September 1995. This date was later moved to November due to programming problems. “This happens,” and the project group, now expanded with one more programmer from the firm, did not take any immediate action. The schedule for implementation was once again changed in mid October. The new date was set to January 1996. This made the users and the analyst worried. Senior management at the firm was contacted and another programmer from the firm was added to the group in the middle of November.

The system was subjected to a first test, performed by the project group, in the beginning of December. ADMIN was very unstable and the group could not test the system in a simulated work situation. The programmer assured that this would be no problem to fix. Two weeks later a new test was scheduled. This time the system behaved the same way as it did in the previous test. Further testing was canceled and the analyst contacted senior management of the consultancy firm. They admitted that the first programmer lacked experience with the used tools. The meeting resulted in a new contract: ADMIN would be ready and implemented in late February 1996.

A third test was performed in late January, but the planned two day test was canceled after two hours. To speed up the work, the consultancy firm assigned two new programmers to the project. A fourth test was performed a week later, and like the previous tests, it failed. The

analyst and the users conducted a smaller audit of the system. The program structure of ADMIN was not satisfactory, but the programmer assured that he would personally re-structure the code. The project manager did not feel very comfortable with the situation. Two weeks later he decided to call in a very experienced “in-house” programmer to do a more extensive audit of ADMIN. The program code was a disaster. Moreover, the expert concluded that ‘Access’ would not work properly in the planned client/server solution. This as well as other circumstances showed that the firm did not have the situation under control. The project manager decided to halt the project. He and senior management decided to do a “restart” of the project, with new and more experienced programmers. The old system should be used as a specification for the new system and Access was to be replaced by another tool. Implementation of ADMIN was scheduled for the last week in May 1996.

6. Discussion

Some might consider evolutionary systems design, feedback cycles, software development as a learning and communication process, etc., as solutions to avoid projects such as ADMIN. But the point made here is that people and organizations can become overcommitted to a failing course of action. Although, ADMIN never became a runaway project it exhibits characteristics consistent with the escalation model.

6.1 Analyzing the Case

Retrospectively, the project manager admitted that there were early warning signals that the project might be in trouble. Analysis of the project shows several escalation factors. (1) ADMIN was the system analyst’s first project as a manager and the interview revealed that it was very important to him that the project succeeded. Through the project, he had an opportunity to improve his position in the organization. The project group was very inexperienced and many decisions were made by the project manager alone. (2) From the very start of the third phase the (first) programmer and the other project members had problems communicating. A number of misunderstandings occurred because of this, but the project manager did not see this as a major problem at first. He believed that the communication between him and the programmer would improve. It did not. Later he realized that he should have acted on this problem earlier. (3) The project manager had the optimistic view that the problems would be solved with time: “Admitting that the project has problems becomes harder as time passes.” (4) The project manager saw a restart of the project as costly. And a restart would mean that earlier investments would be wasted and he would have to admit that mistakes were made. (5) The central systems department selection of tools came to be a major reason for the trouble in ADMIN. As the organization is decentralizing the role and power, of the systems department is changing. (6) The first

programmer was not properly trained in the tool to be used, and this was not admitted by the firm until the project was halted.

6.2 Avoiding Runaway IT Projects

People and organizations can become overcommitted to a failing course of action, getting themselves into so called escalation situations. It is claimed that escalation as a reason for IT project failure is given too little attention in most of the risk management literature and in practice. Escalation situations occur when organizational projects have little salvage value, when decision makers want to justify their own past behavior, when people in a project are bound to each other, and when organizational inertia and internal politics combine to prevent a project from being shut down.

Research suggests that escalation traps or runaway projects can be avoided. These suggestions focus on two aspects: the individual action, i.e., what IT project managers can do and the organizational actions [2]. Here I would like to elaborate these two aspects further and add a third aspect, falling in between the individual and the organizational. A revised framework for avoiding escalation would include:

- **The Aspect of Professionalism.** All IT professionals have ethical duties when it comes to reporting on the status of a project [32-34]. Project managers, and other decision makers, must recognize that there is a natural tendency to escalate when one becomes too committed to a course of action. If project managers are aware of other escalation situations and the forces “driving” to persist on a course of action and “restraining” withdrawal in the situation, their propensity to escalate in the next project is probably lower. It is important for individuals to stay with the overall objective of a project but not with a particular solution since an alternative solution might always be considered.
- **The Aspect of Decision Processes.** The project manager must ensure that as many decisions as possible are subject for discussion, e.g., no decision should be made without explicit consideration of the disadvantages or risks involved in the decision alternative. Negative aspects must be surfaced in all decisions to be made. If no negative aspects are found, postpone the decision until next day or next meeting. Since the final decision to continue a failing project often is made by an individual, the process leading to the decision should be a group effort [21]. For this, conflict is a mechanism for facilitating learning, e.g., by the use of a devil’s advocate, an individual who plays the formal role of critic to help the decision maker test the assumptions and the logic of the ultimate decision. The importance of the group in the decision process is further emphasized since many decisions made in projects concern problems which are not well defined, i.e., soft problems [35], and have to be discussed from many perspectives.

- **The Aspect of Organizational Culture.** Organizations should, to a greater extent, use formal methods to monitor the progress of projects. Serious project audits must be executed on a regular basis. Larger projects have a higher risk for escalation and a greater need for control. These large projects have higher complexity, more stakeholders with different views and criteria for success, greater resource requirements, greater scope and more interactions resulting in more opportunities for inadequacy. Different reactive activities, such as indicators, evaluation, control and assessments are important management issues in project activities. Most organizations have monitoring functions to control deviations in projects, and functions are added to monitor the functions and so on (a common way to control what is not under control). But, no matter how thorough the control, audits and revisions are, all problems are possible to hide, until it is too late to deal with them. Therefore, a proactive approach to avoid escalation is necessary. With an explicit company policy on failure people in the organization have guidelines for how to act in an escalation situation. An attitude such as “we have never abandoned any project in this organization” will surely promote escalation. A central issue in a proactive approach is incentives, such as rewards for project members as well as corrective action when called for. Organizations have to create such an open environment or culture that individuals and groups are forced to raise the questions necessary to avoid project escalation. How to do this is a contingent issue each organization has to work out.

6.3 Recommendations for Education

Curriculums for the training of IT professionals, or whatever we decide to call them, must provide basic knowledge on how we behave and act as humans in a failing situation. We should not only provide the students with holistic methods and “super tools,” but complement them with an awareness and understanding of how people behave, for example, in escalation situations. Today most text books covering IT project management might have a line or two about escalation situations in projects. University courses such as introduction to software engineering, analysis and design, etc., should allocate time for the students to work with cases and discuss failing projects. Our curricula must show that failure is natural for projects and software development and that failure is a prerequisite for innovation.

7. Conclusions

One of the main tasks in projects is to avoid runaway projects. But if organizational members do not know about escalation situations, if the decision procedures have fallacies, and if there is an organization with a culture promoting escalation, we will have runaway projects.

Since our discipline was founded in the 60s, data processing systems have been developed in projects. Today, education focuses on the nature of these projects and the methods used in the projects, but in a longer perspective, some are convinced that these projects will be a less important concept [36]. However, the tendency to escalate will still be an important phenomenon in all human activity. We will still find ourselves in situations where it is hard to decide whether we should continue or stop, saying: "Can we afford to quit or do we have to continue?"

8. Acknowledgments

This research is partly funded by the Swedish National Board for Industrial and Technical Development (NUTEK). I am also grateful to the project manager for sharing his experience with the project and to colleagues for constructive comments.

9. References

- [1] A. I. Teger, *Too Much Invested To Quit*. New York: Pergamon Press, 1980.
- [2] M. Keil, "Pulling the Plug: Software Project Management and the Problem of Project Escalation," *MIS Quarterly*, vol. 19, no. 4, 1995.
- [3] J. Ropponen, "Risk Management in Information Systems Development," Working Paper, *Department of Computer Science and Information Systems*. Jyvaskyla, Finland: University of Jyvaskyla, 1993.
- [4] S. P. Keider, "Why Projects Fail," *Datamation*, December, pp. 53-55, 1974.
- [5] D. McComb and J. Y. Smith, "Systems Project Failure: The Heuristics of Risk," *Journal of Information Systems Management*, pp. 25-34, 1991.
- [6] M. Keil, "Identifying and Preventing Runaway Systems Project," *American Programmer*, vol. 8, no. 3, pp. 16-22, 1995.
- [7] K. Ewusi-Mensah and Z. H. Prazasnyski, "Factors contributing to the abandonment of information systems development projects," *Journal of Information Technology*, vol. 9, pp. 185-201, 1994.
- [8] T. K. Abdel-Hamid and S. E. Madnik, "The Elusive Silver Lining: How We Fail to Learn from Software Development Failures," *Sloan Management Review*, pp. 39-48, 1990.
- [9] C. Sauer, *Why Information Systems Fail: A Case Study Approach*. Alfred Waller, 1993.
- [10] K. Lyytinen and R. Hirschheim, "Information systems failures - A survey and classification of the empirical literature," in *Oxford Surveys in Information Technology*, vol. 4: Oxford University Press, pp. 257-309, 1987.
- [11] E. Oz, "When Professional Standards are Lax - The CONFIRM Failure and its Loss," *Communications of the ACM*, vol. 37, no. 10, pp. 29-36, 1994.
- [12] J. P. Near and M. P. Miceli, "Effective Whistle-Blowing," *Academy of Management Review*, vol. 20, no. 3, pp. 679-708, 1995.
- [13] H. L. Angle and J. L. Perry, "An Empirical Assessment of Organizational Commitment and Organizational Effectiveness," *Administrative Science Quarterly*, vol. 26, pp. 1-13, 1981.

- [14] C. A. Kiesler, *The Psychology of Commitment: Experiments Linking Behavior to Belief*. New York: Academic Press, 1971.
- [15] G. R. Salancik, "Commitment and the Control of Organizational Behavior and Belief," in *New Directions in Organizational Behavior*, B. M. Staw and G. R. Salancik, Eds. Chicago: St. Clair Press, 1977, pp. 1-54.
- [16] B. M. Staw, "Counterforces to Change," in *Change in Organizations*, P. S. Goodman & Associates, Eds. San Francisco, CA: Jossey-Bass, 1982, pp. 87-121.
- [17] B. M. Staw, "The Escalation of Commitment To a Course of Action," *Academy of Management Review*, vol. 6, no. 4, pp. 577-587, 1981.
- [18] B. M. Staw and J. Ross, "Behavior in Escalation Situations: Antecedents, Prototypes, and Solutions," in *Research in Organizational Behavior*, vol. 9, L. L. Cummings and B. M. Staw, Eds. Greenwich, Connecticut, Jai Press Inc., 1987, pp. 39-78.
- [19] J. Brockner, "The Escalation of Commitment to a Falling Course of Action: Toward Theoretical Progress," *Academy of Management Review*, vol. 17, no. 1, pp. 39-61, 1992.
- [20] M. G. Bowen, "The Escalation Phenomenon Reconsidered: Decision Dilemmas or Decision Errors?," *Academy of Management Review*, vol. 12, no. 1, pp. 52-66, 1987.
- [21] G. P. Schneider, "Escalation Behavior in Information Systems Development: Alternative Motivations, Experience, and the Sunk Cost Effect," Dissertation, *College of business Administration.*, The University of Knoxville, Tennessee, 1993.
- [22] S. Nathanson, J. Brockner, D. Brenner, C. Samuelson, M. Countryman, M. Lloyd, and J. Z. Rubin, "Toward the Reduction of Entrapment," *Journal of Applied Social Psychology*, vol. 12, no. 3, pp. 193-208, 1982.
- [23] B. M. Staw and J. Ross, "Knowing when to pull the plug," *Harvard Business Review*, vol. 65, no. 2, pp. 68-74, 1987b.
- [24] R. Sabherwal, M. K. Sein, and G. Marakas, "Why Organizations Increase Commitment to Failing Information Systems Projects?," Working Paper, Department of Decision Science and Information Systems, Florida International University, Miami, November 1994.
- [25] M. H. Bazerman, T. Giuliano, and A. Appleman, "Escalation of Commitment in Individual and Group Decision Making," *Organizational Behavior and Human Performance*, vol. 33, pp. 141-152, 1984.
- [26] D. Kahneman and A. Tversky, "Prospect Theory: An Analysis of Decision Under Risk," *Econometrica*, vol. 47, no. 2, pp. 263-291, 1979.
- [27] H. Garland, "Throwing Good Money After Bad: The Effect of Sunk Cost on the Decision to Escalate Commitment to an Ongoing Project," *Journal of Applied Psychology*, vol. 75, no. 6, pp. 728-731, 1990.
- [28] H. R. Arkes and C. Blumer, "The Psychology of Sunk Cost," *Organizational Behavior and Human Decision Processes*, vol. 35, pp. 124-140, 1985.
- [29] P. D. Harrison and A. Harrell, "Impact of "Adverse Selection" on Managers' Project Evaluation Decisions," *Academy of Management Journal*, vol. 36, no. 3, pp. 635-643, 1993.
- [30] J. Brockner, S. Nathanson, A. Friend, J. Harbeck, C. Samuelson, R. Houser, M. H. Bazerman, and J. Z. Rubin, "The Role of Modeling Processes in the "Knee Deep in the Big Muddy" Phenomenon," *Organizational Behavior and Human Performance*, vol. 33, pp. 77-99, 1984.

- [31] R. H. Hall, *Organizations - Structures, Processes, & Outcomes*. Englewood Cliffs, New Jersey: Prentice Hall Inc., 1991.
- [32] H. J. Smith and M. Keil, "Speaking Out to Outsiders," *Beyond Computing*, July/August, pp. 20-21, 1995.
- [33] H. J. Smith and M. Keil, "Mum's the Word," *Beyond Computing*, June, pp. 16-17, 1995.
- [34] R. E. Anderson, D. G. Johnson, D. Gotterbarn, and J. Perrole, "Using the New ACM code of ethics in decision making," *Communications of the ACM*, vol. 36, no. 2, pp. 98-107, 1993.
- [35] P. Checkland, *Systems Thinking, Systems Practice*: John Wiley & Sons Ltd, 1981.
- [36] B. Dahlbom, "Goteborg Informatics," *Scandinavian Journal of Information Systems*, vol. 7, no. 2, pp. 87-92, 1995.