Shaun Goldfinch
University of Otago, New Zealand

Pessimism, Computer Failure, and Information Systems Development in the Public Sector

Shaun Goldfinch is a senior lecturer in the Department of Political Studies at the University of Otago, New Zealand, and Associate Professor of Public Administration, American University of Sharjah, UAE. He is the author of *Remaking New Zealand and Australian Economic Policy* (Georgetown University Press, 2000).
E-mail: shaun.goldfinch@stonebow.otago.ac.nz

The majority of information systems developments are unsuccessful. The larger the development, the more likely it will be unsuccessful. Despite the persistence of this problem for decades and the expenditure of vast sums of money, computer failure has received surprisingly little attention in the public administration literature. This article outlines the problems of enthusiasm and the problems of control, as well as the overwhelming complexity, that make the failure of large developments almost inevitable. Rather than the positive view found in much of the public administration literature, the author suggests a pessimism when it comes to information systems development. Aims for information technology should be modest ones, and in many cases, the risks, uncertainties, and probability of failure mean that new investments in technology are not justified. The author argues for a public official as a recalcitrant, suspicious, and skeptical adopter of IT.

The majority of information systems (IS) developments are unsuccessfu1.[1] The larger the development, the more likely it will be unsuccessful. Though the exact numbers are uncertain and depend to some extent on how success is measured, something like 20 percent to 30 percent of all developments are total failures in which projects are abandoned. Around 30 percent to 60 percent are partial failures in which there are time and cost overruns or other problems. The minority are those counted as successes (Collins and Bicknell 1997; Corner and Hinton 2002; Georgiadou 2003; Heeks and Bhatnagar 1999; Heeks 2002, 2004; Iacovou 1999; James 1997; Korac-Boisvert and Kouzmin 1995; Standish Group 2001, 2004).

A U.S. survey of IS projects conducted by the Standish Group in 2001 found that success rates varied from 59 percent in the retail sector to 32 percent in the financial sector, 27 percent in manufacturing, and 18 percent in government. Overall, the success rate was 26 percent. In all, 46 percent of projects had problems, including being over budget and behind schedule or being delivered with fewer functions and features than originally specified. Another 28 percent failed altogether or were cancelled. Cost overruns averaged nearly 200 percent. This success rate varied dramatically by total project budget: For projects under US$750,000 the success rate was 55 percent; for those with budgets over US$10 million, no projects were successful (SIMPL/NZIER 2000). More recent Standish Group (2004) estimates saw a success rate of 29 percent, but 53 percent of projects had problems and 18 percent failed.

A New Zealand government study judged 38 percent of government projects a success, while 59 percent involved problems and 3 percent were a complete failure or were cancelled. Government success rates, at 31 percent, were slightly higher than private sector success rates. Above the NZ$10 million mark, however, the success rate for both was *zero* (SIMPL/NZIER 2000). One study of hundreds of corporate software developments found that five out of six projects were considered unsuccessful, with one-third cancelled outright. Of the two-thirds that were not cancelled, price and completion times were almost twice what had originally been planned (Georgiadou 2003). The Royal Academy of Engineering and the British Computer Society (2004) found that 84 percent of public sector projects resulted in failure of some sort.

The sums involved in such projects can be staggering. A study of IS developments in the British public sector estimated that 20 percent of expenditures were wasted, and a further 30 percent to 40 percent led to no perceivable benefits (Wilcocks 1994). In 1994, the U.S. General Accounting Office reported that spending of more than US$200 billion in the previous 12 years had led to few meaningful returns.

Collins and Bicknell (1997) estimate that public sector failures in the United Kingdom cost £5 billion. For example, the Resource Management Initiative in the U.K. National Health Service led to new information systems being introduced in almost every

hospital. Despite the expenditure of hundreds of millions of pounds, few were successful "by any criteria" (Heeks and Bhatnagar 1999, 59). The Wessex Health Authority's Regional Information Systems Plan was cancelled after £43 million–£60 million pounds (the actual figure was uncertain) had already been spent, with almost nothing achieved (Collins and Bicknell 1997).

After years of development provided little more than an e-mail system and a number of terminals to a 1970s-era mainframe, the New Zealand Police abandoned an IS development in 1999, at a cost of more than $NZ100 million—around one-eighth of the yearly budget. A benefit payment scheme involving the British Post Office, the Department of Social Security, and the computer company ICL was abandoned after three years at a cost of £300 million (*The Economist* 2002). An already obsolete air-traffic support system opened in Swanson, United Kingdom, in 2002, six years late and £180 million over budget (*The Economist* 2002). The Canadian Firearms Program blew out from initial estimates of C$113 million to more than C$1 billion, an overrun of almost 900 percent (Auditor General of Canada 2002).

Vast sums of money, mostly provided by aid agencies, have been spent on health and other information systems in South Africa, on donor-funded IS projects in China, and on World Bank–funded projects in Africa, nearly all of which have been total or partial failures (Heeks 2002, 2004). Spectacularly, the U.S. Internal Revenue Service, with an annual computer budget of US$8 billion, managed "a string of project failures that have cost taxpayers $50 billion a year [mainly defined as revenue forgone]—roughly as much as the yearly net profit of the entire computer industry" (James 1997, 1). Around US$150 billion is wasted per annum on information and communications technology failures in the public and private sectors in the United States and US$140 billion in the European Union (Dalcher and Genus 2003). Despite the catalogue of continuing failures, and despite decades of attempts to deal with the problem, there is little evidence that IS developments failures are decreasing in the public sector.

Given its ubiquity and the vast sums involved, computer failure has received remarkably limited coverage in the public administration literature. Indeed, as Northrop (2003) points out, notwithstanding their centrality to the public sector, information technology (IT) issues receive surprising little attention. In general, when they are discussed, there is often an underlying assumption that new information technology is more or less a "good thing." This article departs from the overwhelming positive view of IT in much of the public administration literature. In the face of persistent and pervasive failure and this enthusiasm for IT, I argue instead for a pessimism in IS development.

...computer failure has received remarkably limited coverage in public administration literature.

The failure of large and complex IS developments is largely unavoidable. Even systems that work in a technical sense often do not deliver all that is expected of them. This is not a problem simply to be solved by a new management or software engineering technique or some other silver bullet, as much of the literature suggests. Rather, the central problem is the overblown and unrealistic expectations that many have regarding information technology. In this article, a model of the four enthusiasms that encourage these overblown expectations is outlined. Once begun, highly complex projects are extremely difficult to monitor and control and to prevent from failing. When projects fail, it is difficult to find and hold to account those responsible. As such, investments in technology should be approached with great caution in the public sector. Large and complex developments in information technology, particularly if new technology is involved, should be avoided if at all possible. If investments are to be made at all, they should have modest aims and use proven technologies. In many cases, it may be better to avoid an investment altogether and continue with existing systems, at least until the technology improves. In sum, rather than the entrepreneurial risk-taking public manager often found in some New Public Management literature, this article argues for a public official as a recalcitrant, suspicious, and skeptical adopter of IT.

Information Systems Failure

Failure is a social construct, and perceptions of what is and is not failure can vary between persons and over time (Bovens and 't Hart 1996). Opinions differ as to whether failure is a normal part of public policy or whether it is an unusual—if sometimes spectacular—event seized on by the media for the sake of eye-catching headlines (Bovens, 't Hart, and Peters 2001). What distinguishes IS failure from other failures in the public sector, however, is its overwhelming ubiquity. Indeed, though some writers might claim that failure in IS development is overstated, the bulk of the research suggests that failure might even be the norm. As Mahaney and Lederer note, "because this problem has endured for three decades, many IS professionals have accepted failure as inevitable" (1999, 291).

Bascarini (1999) notes that "a standardized definition of project success does not exist, nor [is there] an accepted methodology for measuring it." Indeed, what is counted as failure depends on who you ask, and perceptions of success and

failure may change over time (Larsen and Myers 1999; Wilson and Howcroft 2002). The Standish Group sees success in narrow terms if a project is delivered on time and on budget, with functions and features delivered as originally specified. The firm KPMG defines "runaway projects" as those that overrun their projected budget or completion date by more than 30 percent, whereas others propose an overrun on budget or time frame of 100 percent as a measure of failure (Cole 1995; Glass 1998). However, failure does not necessarily imply only technical failure—that is, even if a system performs as its designers intend, it may not be used as intended, or used at all, and so still be considered at a failure (Dutton et al. 1995; Laudon and Laudon 1998). Even projects that meet design specifications may not increase worker productivity or deliver other gains expected; productivity may even decrease.

Indeed, there is considerable debate regarding productivity and other benefits of IS in the last three decades, in both the public and private sectors and in the economy generally (Brown and Brudney 2003; Coltman et al. 2001; Hewson and Hewson 1998; Holden 2003; Kraemer and Dedrick 1994; Norris and Moon 2005). The benefits may not offset the costs of development; a long project can cause years of serious and costly disruption to operations that may never be recovered (Norris and Moon 2005; Teega Associates 2003). Alternatively, despite technical flaws such as project overruns, overspending, design questions, and so on, some systems may be deemed a success (Wilson and Howcroft 2002). A project may initially be considered a failure, only later to be reevaluated as a success for reasons quite outside the technical success of the project (Huang et al. 2003).

Drawing on Wilson and Howcroft, a number of types of failure can be summarized:

- **Project failure:** The project does not meet the standards agreed, including the functions provided, budget, or completion deadlines (Wilson and Howcroft 2002, 237).
- **System failure:** The system does not work properly, including not performing as expected, not being operational at the specified time, or not being used in the way intended (Wilson and Howcroft 2002, 237). Even when used as intended, the project may not generate productivity gains or deliver the benefits expected.
- **User failure:** The system is not used in the face of user resistance because of such things as recalcitrance, lack of training and ability of staff, and the complexity of the new system.

Why Do Information Systems Projects Fail?

There is a considerable literature on the causes of IS failure and the solutions to this ongoing problem, largely reflecting the latest management fad. None has been particularly successful in preventing failures, and most have been abandoned for new solutions. Of the more promising solutions, Heeks has proposed a checklist of "critical failure factors" (Heeks 1999; Heeks and Bhatnagar 1999). These include data inadequacies; technical problems; management, process, and technical skill shortages; cultural clashes and political infighting; and external environmental factors (see table 1).

Though useful, the factor approach has been critiqued by a number of writers. First, "the factor approach tends to view implementation as a static process instead of a dynamic phenomenon, and ignores the potential for a factor to have varying levels of importance at different stages of the implementation process" (Larsen and Myers 1999, 398). Second, the relationship among the factors is often unexplained (Ginzberg 1981; Lucas 1981). Indeed, the approach assumes that each factor is an independent variable and underplays the interaction among them (Bussen and Myers 1997; Nandhakumar 1996a). Third, a number of studies have shown a lack of consistency in the importance of factors, and few have been important in all cases (Kwon and Zmud 1987). Fourth, the factor approach is claimed, perhaps unfairly, to be overly mechanist while underplaying the importance of such things as organizational culture in development and the importance of the political, social, and environmental contexts, both within and outside the organization (Bussen and Myers 1997; Nandhakumar

Table 1 Critical Failure Factors in IS Disasters[1]

Factor	Description	Factor	Description
Information	Information and data inadequacies	Cultural	Clashes with national/local culture
Technical	Problems with IT such as incompatibility across agencies	Structural	IS clashes with organizational and/or management structures
People	Lack of staff with sufficient training, skills or inclination to handle or develop IT	Strategic	IS not coordinated across different agencies or divisions
Management	Lack of management skills, knowledge and training	Political	Political infighting derails project
Process	Processes are inadequate to integrate community or channel relevant information	Environmental	Factors outside the organization disrupt project

Source: Heeks and Bhatnagar (1999).

1996b). This last criticism would be difficult to sustain in relation to Heeks's critical factor approach, which explicitly takes into account such environmental, cultural, and political factors. Finally, the critical failure factor, while possibly useful in explaining failure in retrospect, provides little guidance how to avoid it. Indeed, if anything, the approach highlights the remarkable complexity of IS developments, the myriad problems that can occur, and the difficulty of addressing them. And it does not adequately address why large projects continue to be initiated, as I will do here.

More recently, IS failure has been seen as a problem of learning where organizations fail to learn from external and internal sources and where they fail to moderate beliefs and behavior in the face of failure. This includes persisting in IS developments long after they have proved to be unfeasible (Irani, Sharif, and Love 2001; Lyytinen and Robey 1999; Wastell 1999). Organizations fail to learn, it is claimed, because of limits imposed by such things as information overload, high turnover of skilled staff, embedded ways of thinking, the drawing of strong conclusions from limited individual experiences, and an inability (and lack of incentives) to draw lessons from previous failure (which are often forgotten in any event) (Lyytinen and Robey 1999).

The separation of IS from other units within a business and the overwhelming technological and engineering background required can also isolate IT specialists from wider issues and learning. Over time, it is claimed, organizations "accept and expect poor performance while creating organizational myths that perpetuate short-term optimization" (Lyytinen and Robey 1999, 85). The high levels of stress and anxiety involved for participants in IS projects also undermine learning (Wastell 1999). Groups and individuals cope with this stress through defense-avoidance behavior patterns, such as reliance on organizational rituals, political infighting, and isolationism, in which groups develop a laager mentality and an inward-looking, mutually supporting, and outwardly suspicious groupthink ('t Hart 1994). These defenses work to avoid having to engage with the problems in the project and to face up to and learn from these problems (Wastell 1999). Failure then, it is claimed, can be reduced by creating incentives for organization and individuals to learn, such as rewarding the bearers of bad news and mistakes, providing a supportive environment for learning, integrating IS into the wider organization, and broadening the focus of IS developers away from the mainly technical (Lyytinen and Robey 1999, 85; Wastell 1999).

Although it provides some insights, the "learning organization" has been seen as having questionable empirical foundations (Salaman 2001). It is unclear as to what learning actually is and who (the organization or the individual) is carrying out the learning—and sometimes the two are conflated in any event (Popper and Lipshitz 2000). It underplays the barriers that exist to learning because of the strong culture, hierarchy, and rituals of organizations and the paradigms, discourses, or belief systems, mostly external to the organization, within which managers work and that rule out some options and constrain others (Salaman 2001). As March and Olsen (1996) argue, there are certain things that can and cannot done—that is, there is a certain "logic of appropriateness" that can provide barriers to learning.

The question also arises how useful some of the learning organization literature is to explaining IS failures, particularly in the public sector. First, the continued high level of IS failure shows there is little evidence of learning occurring, now or in the future. Second, it is unclear what the organization should learn in order to avoid IS failure. There is no agreement on which software development methodologies or latest management systems are effective. None has been particularly successful so far. A number of critics note that management is characterized by the adoption of fads whose popularity has little to do with actual evidence of effectiveness. These fads are usually abandoned after a few years in the face of less than fantastic results before a new fad is embraced equally as enthusiastically (Birnum 2000; Brindle and Stearns 2001; Connor 1997). Systems for government information technology management show some of the same tendencies (cf. Holden 2003). Software development is similarly afflicted. As Georgiadou notes,

> Researchers and practitioners endeavoured to find ways of improving the productivity and quality of products. New languages were often believed to have almost magical powers of getting over the crisis. Automated tools, formal methods and, more recently, the object-orientated approach were proposed as alternative "religions" [with] software engineers becoming ardent followers of one approach or another.
>
> Computer science, information systems, and software engineering created a large number of lifecycle models, methodologies, and metrics designed for solving problems of quality of both software products and software processes. Unfortunately, only few of them were subjected to careful experimentation and were often adopted with limited empirical evidence of their correctness or effectiveness. (Georgiadou 2003, 139)

Despite decades of development, there are still no reliable methods for estimating project size or costs and no software development methodology that can guarantee good results (Georgiadou 2003).

Software estimation still relies on the judgment of skilled individuals for success, and even the best can be wrong (Evans and Reynolds 2001). Fenton and Pfleeger (1997) found that experienced managers were more than 50 percent wrong in estimating software costs, and some software tools produced results that were incorrect by 200 percent. This was for completely specified projects, using experienced people and well-proven technology. Jacobson, Rumbaugh, and Booch found that the estimated project size at the beginning of a project "may differ from the final size by a substantial percentage, say 50 percent" (1998, 354).

The learning literature focuses on IS within a firm or organization and integrating that IS division or section into the wider needs of the organization. However, government agencies often depend on external consultants to advise them on IS developments and on large IT companies to develop and supply software and hardware. Failure might encourage learning, but it may not have any great benefit for the purchaser. A consultancy company might learn that advising a client to proceed with a large IS project guarantees a flow of income in the future, quite apart from the value that this IS project provides to the purchaser and with little risk that poor quality of advice will affect financial rewards then and in the future. Such IT companies and their salespeople might learn that promising largely unachievable results from highly expensive IS projects is a good way to subsidize their software and hardware research and add to the company's profits, with little risk that contracts, however many pages they might run to, will be able to hold them to account. There is little evidence that consultants, IT companies, public agencies, or many practitioners and academics have learned one of the key lessons of IS failure—that large and ambitious projects should be treated with great caution or avoided altogether.

Problems of Enthusiasm

Enthusiasm for large and complex investments in IS continues unabated despite decades of failure. Indeed, the largest-ever public sector project was initiated in 2002 by the United Kingdom's National Health Service at an estimated cost of US$11 billion. To explain why large and ambitious projects continue to be initiated, I propose a model containing four pathological enthusiasms, each of which is linked to a key player or group within public sector IS developments.

Enthusiasm for large and complex investments in IS continues unabated despite decades of failure.

The first enthusiasm is *idolization,* or technological infatuation, in which public officials "use computers and are overaware of IT's potential. They believe that IT can transform the business of government. The public sector becomes awash with IT driven reform projects, which place technology at the heart of the change process" (Heeks and Davies 1999, 27). Indeed, public servants can get carried away with the excitement of it all, providing reports and projections for the benefits of new developments that verge on the fantastic (Dale and Goldfinch 2002).

The second enthusiasm is *technophilia,* or "the myth of the technological fix," in which "the entire IS profession perpetuates the myth that better technology, and more of it, are the remedies for practical problems" (Lyytinen and Robey 1999, 95). Many of those entering the IT industries are, in common parlance, geeks: They are enthusiasts for computers and technology, excited by the possibilities that new technologies offer and by the challenging intellectual puzzles that developing new technology brings. Technological development can become an end in itself. And, as noted, programmers are also subject to bouts of enthusiasm for the new programming methodologies that come along every few years despite little evidence of their efficacy (Georgiadou 2003).

The third enthusiasm is what I call *lomanism,* drawing on Arthur Miller's archetypal salesman Willie Loman in *Death of a Salesman.* Lomanism is the enthusiasm, feigned or genuine, that sales representatives and other employees develop for their company's products and skills and its ability to develop new products and technologies, whatever the objections or questions put forward by potential and actual purchasers and others involved in purchasing and developing the technology. IT salespeople can be faced by an unusually responsive audience; often those involved in finally deciding which systems to buy are those responsible for promoting new developments in the first place. Salespeople or company employees with the temerity to suggest that purchasers' expectations might be somewhat overblown are likely to find that those purchasers will simply go to a company that promises to meet their expectations.

The fourth enthusiasm is *managerial faddism*. This is the tendency of consultants and managers to eagerly embrace the newest management fad, methodology, or uttering of the management guru of the moment and to see problems as largely solvable (or prevented) through better or more "rational" management and the appointment of skilled managers. Such managerial faddism is also reflected in the belief that most problems can be fixed or prevented—and benefits created—by improving management structures along the lines of the new fad, of which new IS projects are often a key element. The orthodoxy of the New Public Management, with its inherent belief in the supremacy of the private over the public sector and its

innovative, entrepreneurial focus, provides a ready ground for such faddism. The public sector must now compete with the private in terms of its adoption of new technologies, including management ones, or face being seen as behind the times and resistant to change. Indeed, one of the benefits of pre-New Public Management Weberian models may have been lost—the much-maligned preference for precedent, stability, and tradition (Du Gay 2000).

Together, these four enthusiasms feed on and mutually reinforce one another in a vicious cycle, creating a strongly held belief that newer and larger IS projects are a good idea. Doubters and skeptics may be portrayed as "negative," "not team players," "not helpful," or, particularly in a public sector influenced by New Public Management and economic models of behavior such as public choice, as "vested" or "rent-seeking" interest groups. Together, these pathologies make up the four enthusiasms of IT failure (see figure 1). When a project does encounter difficulties, these four enthusiasms can undermine attempts to curtail or abandon the project—a project can always be fixed with better management, a redesigned monitoring system or contract, more technology or hardware, better programming, or just a reassuring "it'll be right on the night."

Problems of Control

Rational models of management and decision making seek to impose order and control on a complex and turbulent world that (it is believed) can be more or less understood (cf. Brown and Brudney 2003; Lindblom 1959). However, rather than showing aspects of linear logic, in which cause and effect are known and can be controlled, IS developments show aspects of complex systems in which the "ability to predict the course of events is limited" and in which even small events can lead to unpredictable outcomes (Turner 1998, 1). Much of the writing on IS developments and IS failure suffers from a hubristic belief that once the correct information is available, the right management system and programming methodology adopted, and rational optimizing individuals given the right incentives, the problem of failure will largely be solved. In contrast to this belief, I argue because of problems of agency, immense complexity, and the interaction of human beings of, at best, only bounded or even limited rationality, it is difficult to understand and control large IS developments. It is difficult to recognize and monitor problems, find solutions to these problems, and hold people accountable for failures. The sheer complexity of IS developments means that humans with not-unlimited abilities are faced with informational overload. Public agencies also work within a sometimes unstable legislative environment, with computer requirements subject to change in the face of changing laws, only further adding to complexity and cost (Small 2000). I will expand on these issues in the next section.

Problems of Agency and Information

There can be considerable problems of agency in IS developments. Agency theory focuses on the problems faced by the principal (such as the manager or chief executive) in controlling an optimizing agent (such as a programmer, IS developer, or project manager) in a situation of information asymmetry and the difficulty of monitoring when the agent may have greater knowledge of the problems and an incentive to conceal them. In many cases, reliable information on the actual progress of projects (as opposed to reported progress) is simply not available because of the intangible nature of software development, the often dynamic and changeable nature of the project itself, the large number of diverse participants involved in the projects who may

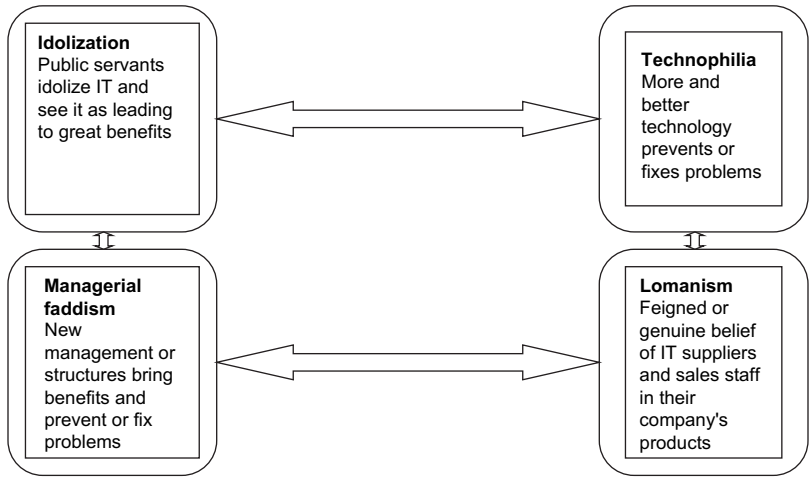

Figure 1 The Four Enthusiasms of IS Failure

be working in different parts of the world, and the sheer complexity and information overload faced in a large project (Smith, Keil, and Depledge 2001). Reflecting this, in many cases, progress and audit reports are largely exercises in hope and fantasy (Dale and Goldfinch 2002). When problems are apparent, they may not be made known to management or monitoring agencies. That members of organizations are reluctant to be the bearers of bad news is a well-reported phenomenon (Tesser and Rosen 1972).

IS developments are no different, with bad news often underreported, concealed, and sometimes even falsified (Collins and Bicknell 1997; Dale and Goldfinch 2002; Heeks 1999; Smith, Keil, and Depledge 2001). In some case, public agencies have been reluctant to give information to monitoring agencies because of claims of "commercial sensitivity" (Small 2000). Even if bad news is reported, it may not be listened to; indeed, bearers of bad news can suffer sanctions themselves (Keil and Robey 2001). When a senior manager or chief executive is tightly linked to the project and identified with its success, he or she may be reluctant to admit to bad news or to curtail the project in the face of difficulties (Collins and Bicknell 1997; Dale and Goldfinch 2002; Heeks 1999; Keil and Robey 2001; Smith, Keil, and Depledge 2001). In any event, managers may lack the ability to evaluate projects, and there may be a tension between authority from expertise and authority from position, particularly in a highly complex fields such as IS development, where management may be supervising people with highly specialist skills that these managers do not necessarily understand (Beetham 1996). Appointments in bureaucracies are often made on the basis of seniority and successful political behavior rather than simply merit, which, again, can be dangerous in highly complex fields such as IS development. Management may simply be afraid of asking "stupid" questions for fear of losing face (Collins and Bicknell 1987).

Even without deliberate distortion, there may be miscommunication and misunderstandings and a considerable degree of tension between different players in public sector IS developments. Professional groups have their own languages, their own ways of doing things, and their own understanding of the world—what is generally called a "culture." Bureaucratic culture can sit uncomfortably with the individualistic, heroic culture of the programmer and the faddish culture of the management consultant and New Public Management–influenced managers (see figure 2). For example, though some writers have alluded to the conservatism of IT engineers and their willingness to downplay what they can achieve (Glass 1999), most note the individualistic and heroic nature of the programming culture, in which difficulties and possible failures are just further challenges to be solved by hugely talented programmers (Bronson 1999; Swanson 1988). This particular technological focus of many IT and IS specialists adds to the inattentiveness to problems of failure and the appropriateness of technology to the organization and, indeed, the resentment of and resistance to management, personal, and political factors and reporting requirements that might interfere with the technological puzzle at hand. As Lyytinen and Robey note,

> The profession of ISD is characterized by specialized training and circumscribed theorizing. Since the dawn of business computing, training in IS has meant 'computer training,' and IS professionals remain technologists at heart. Unfortunately, a technologist's perspective does not encourage an accurate diagnosis of the role of computing in business strategy and operations. (1999, 94)

Indeed, the different professional values, modes of behavior, language, and belief systems of different participants in IS developments can lead to confusion,

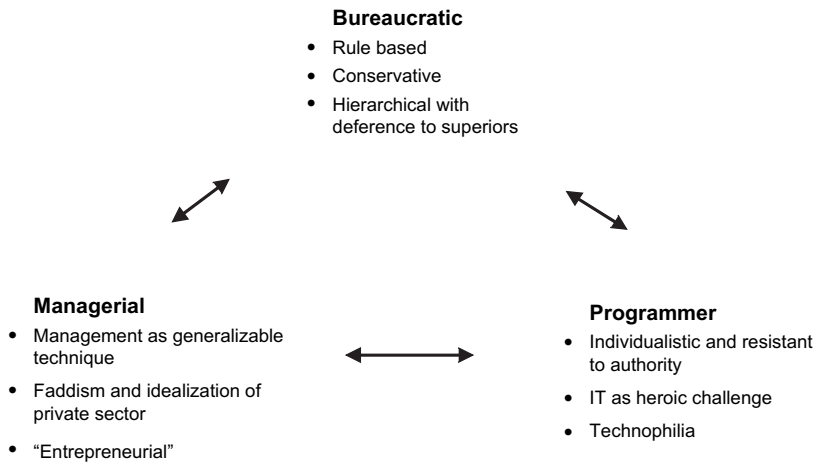

Figure 2 Culture Clashes in Public Sector IS Developments

incommensurability, and a degree of "talking past" one another. The risk-taking entrepreneurial culture of managers and managerial consultants—and their often baffling jargon—may add to the confusion and danger of failure, especially as New Public Management models and New Public Management–influenced managers and consultants often sit atop or beside largely traditional bureaucracies. Line management may find that simply ordering about an individualistic, highly skilled, and somewhat recalcitrant programmer may not have the desired effects. There can be considerable professional jealousies and turf protection, such that attempts to curtail and direct other players are seen as unwarranted interference and are resisted in various ways, including the restriction of information. Indeed, like any situation involving humans, IS developments involve political battles, struggles for individual autonomy, power and value dominance, career advancement, personality clashes, and so on (Grover, Lederer, and Sabherwal (1998); Knights and Murray 1994). The complexities of technological development are only further complicated by the complexities of human relations.

Even when problems are acknowledged, projects may continue because the forces encouraging abandonment are overpowered by the forces encouraging continuance—including a strong belief that "it will be right on the night." In particular, what is called the "completion effect"—the nearness to the successful completion of the project—is a strong driving force. In IS development, this is particularly relevant because of the "90 percent completion" syndrome: The proportion of the completed project increases to the point at which it is estimated to be 90 percent complete, and increases very slowly thereafter (Abel-Hamid 1988). In some cases, projects are reported to be 90 percent complete for half the duration of the project—an obvious impossibility. In a survey of IT auditors, it was found that the completion effect classified more than 70 percent of runaway projects (Keil, Mann, and Rai 2000).

Frontline staff can provide a reality check on the overreaching ambitions of IS developers and management. However, critiques of bureaucratic structures have noted the tendency for members to follow commands even if they are misguided and for responsibility and initiative to be discouraged (Merton 1957). In any event, frontline grumbles may not treated as carefully as they should, especially if the hierarchical culture sits (however uneasily) with public choice and New Public Management notions of management superiority and the treatment of professional groups as rent-seeking interest groups. Management hostility to even mild and, in retrospect, highly justified reservations that frontline staff expressed to IS developments has been noted in failures (Dale and Goldfinch 2002). On the other hand, a factor in failures can be the reluctance or inability of end users to adopt and adapt to the new technology: The benefits of using the system are not seen as justifying the effort to relearn skills so as to use the system, or in some cases, end users actively subvert the computerization process (Al-Gahtani and King 1999; Collins and Bicknell 1997; Dinsdale 2004; Rocheleau 2003; Wilson and Howcroft 2002). As such, developing systems without the involvement and, at very least, tacit approval of staff can be a high-risk proposition.

Contracting Out

The popularity of New Public Management models has partly been a response to the perceived failings and rigidity of the bureaucratic model, as has the related growth of contracting out. In many—if not most—cases, public sector IS developments are outsourced to private companies rather than developed in-house. Management and other consultancies are often hired to give advice on IT and management system requirements, to recommend purchasing arrangements, and to monitor and audit IS developments. However, there is considerable evidence that contracts are not effective at controlling projects and providing sanctions when IS projects fail and that outsourcing has costs and dangers of its own. In the highly complex and changing environment of IS development, no contract, even if it runs for thousands of pages, can foresee and control all aspects of an IS development (Collins and Bicknell 1997; Dale and Goldfinch 2002; Dizard 2001). The complexity and uncertain causes of failure (if and when it does appear), as well as the often remarkably complex contracts that have been developed in (a usually vain) attempt to control the development and provide sanctions, mean that in the face of failure, litigation can be costly and results uncertain. In some cases, it is even suspected that IT companies will build penalty payments for late delivery of applications into their costs when tendering for contracts (Teega Associates 2003).

A number of studies caution against overreliance on external consultants and information technology outsourcing. As Grant notes,

> … intended to save costs and develop competencies, IT outsourcing can be a double-edged sword if there is sufficient internal capacity for performing effective due diligence in IT systems procurement, designing and managing the outsourcing relationship and capturing and embedding new knowledge gained into the organisation. (2003, 173)

Indeed, as Brown and Brudney found, local governments with "higher levels of contracting . . . *were less likely* to have the project delivered on time and within budget and achieved less benefit from the technology in

productivity and performance, organization decision making, and customer service" (1998, 341; emphasis added). In many cases, public sectors do not have the capacity, resources, and personnel to adequately develop and monitor outsourced projects, particularly as during the privatization drives of the 1980s and 1990s, government-owned computer and information technology agencies were often sold off. Higher levels of contracting can also impede the development of this capacity (Brown and Brudney 1998). In a number of countries, particularly developing ones, this lack of state capacity can be pronounced. This leaves the potential for consultant and/or producer capture and for exploitative relationships between often large and powerful multinational IT and consultancy companies and comparatively less powerful and less competent governments; again, this risk is comparatively greater in developing countries with limited state capacity.

Nor does competition provide a solution. In a marketplace, there may be only a few large IT and consultancy companies. As such, contestability may not be sufficient to provide a check on potential capture by allowing competition during the tendering of contracts and by allowing different companies to advise on purchasing; to develop, implement, and maintain different parts of the system; and to monitor and audit projects. In some cases, companies have absorbed rivals involved in developing different parts of a system during the process of IS development, further reducing competition and contestability and increasing the dangers of producer capture (Teega Associates 2003). In any event, external consultants and suppliers are prone to the enthusiasms outlined earlier.

Problems of Control: Conclusion

In sum, large IS projects are extremely difficult to monitor. Is difficult to know whether things are going well or badly until ultimate abandonment and failure. When things go wrong, it is difficult to find those responsible and hold them accountable, even in the unlikely event that any one identifiable person or group is responsible. Even when problems are known, participants may be reluctant to curtail or abandon failing projects and may continue to throw good money after bad. At best, a learning organization, clearer lines of accountability, better managers, tighter milestones for delivery of applications, or other management innovations may make it easier for senior management to find out that a project is off the tracks. But even then, this bad news may not be wanted or heard and may not lead to termination or modification of the project. No one has yet designed a contract that can control and contain all projects. The danger of producer and consultancy capture remains great, particularly as there may be little contestability of advice and supply and a dearth of IS development skills in the public sector. Management improvements will not necessarily stop, for the reasons outlined, ambitious and largely unachievable projects from being initiated and large amounts of money from being spent before a project's ultimate failure and abandonment.

Pessimism as an Information Systems Management Tool

What, then, is to be done? Much of the writing on IS failure suffers from what Oakeshott (1962) might term "rationalism," that is, a belief that there is some technique that can be applied to IS developments that will fix them once and for all. Failure is still so common because that right technique—a new programming methodology, a better designed contract, a new management fad, a new consultancy template, or whatever—has not yet been discovered. Or if it has, it has not yet been applied. In the face of such optimism, and the problems of enthusiasm and control that I have outlined, I suggest pessimism when it comes to IS development. This is a belief that the processes involved in IS development are not fully understood, that their complexity makes them difficult (if not impossible) to control, and that large IS developments are likely to fail. Rather than simply a technical exercise of software engineering, the application of a rational management science and the talents of brilliant managers, or the bringing in of highly skilled consultants and private companies—or even just some healthy combination of all of these—IS developments in the public sector are a potent mix of dangerous enthusiasms, unclear aims and technical specifications; highly challenging technical problems; problems of agency; frailties of humans and management systems; personality and other conflicts; immensely complex contracts; producer and consultancy capture; legislative instability; and clashes of cultures among public servants, software developers, consultants, salespeople, and their respective organizations, among other things. If one thing characterizes large IS developments, it is this incredible complexity. That these developments are often combined with other large-scale reorganizations, and that technical and other specification changes are sometimes made during the development process, only compounds this complexity. It is not likely that some technique or techniques will be able to manage this complexity in the foreseeable future. Indeed, as expectations continue to grow regarding IT and IS, it may be that any improvements made in techniques will be outpaced by the increasing demands made by complexity on all-too-human managers, public servants, software designers, and other members of the IS industry.

At its most basic level, pessimism raises the question of whether new IS developments are actually of benefit. In the garbage can of public management decision making, it is often the solution—which might be

restructuring or investment in IS or both—that exists quite separate from any identified problem (Cohen, March, and Olsen 1972). If the first step, then, is to ask what is to be achieved and what problem is to be solved, the next question becomes, can this problem be solved without investment in further IS? Can some minor adjustment to current systems of management or IS deliver benefits without great costs and great disruption? Can some upgrade to present systems be carried out? If the costs, uncertainties and risks of change and failure are great, do they potentially outweigh the inconveniences, costs, and difficulties of continuing with the way things are already done?

If the decision is made to proceed with some IS development, then the question becomes, how this can be done with the least disruption, the least cost, and the least risk and uncertainty? The most dangerous course would be to invest in high-risk, highly ambitious "bleeding edge" developments with a long development time frame and a very high probability of failure.[2] A more sensible solution would be to examine what is currently working in the marketplace and buy something off the shelf that has been demonstrated to work. However, once adaptations are attempted on an existing system—however well it might work in its home country or public sector—the probability of failure again becomes large (Collins and Bicknell 1997; Heeks 1999). If a system is not in existence anywhere else, then the question should be asked again: Are the expectations for IS development too high? Can expectations be scaled back to what does exist, or can the organization get by without further IS development, at least until the technology improves? In many cases, a few years longer with an obsolete system won't make much of a perceivable difference. It is unlikely that the public sector is going to be competed out of existence by some "first mover" in technology, even if this is the case in the private sector. If the answer is still that an IS development is needed, then—and only then—should an IS development be considered.

Once the decision is made to proceed with an IS development, then again, pessimism should be the guiding principle. Be skeptical about what can be achieved by IS developments. Benefits and successes are more likely when only routine applications are automated (Northrop et al. 1990; Teega Associates 2003). Be aware of the dangers of producer and consultancy capture and the undermining of public sector capacity in IT. Believe that developments will work only when they can be shown to work. The promises and enthusiasms of internal and external IS people, salespeople, consultants, management gurus, technicians, and software engineers are unlikely to be worth much when a system fails, nor will any contract, even if it runs to thousands of pages. If there has to be restructuring of the organization, this should not be tied to the IS development but carried out separately and before. It is unwise to increase complexity by changing specifications during the development. Projects with long development time frames are likely to be overtaken by technology improvements and become obsolete by the time they are operational, and they are more likely to face legislative changes during their lifetime (Teega Associates 2003). Be prepared to terminate the development if cost overruns, delays, or nondelivery become apparent despite claims that it is "almost there" or that "it will be right on the night." The system, if it does work, is unlikely to lead to huge job or cost savings in the short term—despite the many promises and projections otherwise—and there may be a loss of productivity until staff become used to its idiosyncrasies. Excluding frontline staff from development is risky, and they might undermine the system even if it does work. Above all, be pessimistic about information technology.

Be skeptical about what can be achieved by IS developments.

Notes

1. An information system (IS) is a computer system combined with the organization and personnel to produce useful outcomes, as opposed to information technology (IT), which is a general term for the computers, networking, and software used in an information system. Another term often used, particularly more recently, is Information and Communication Technology (ICT). It is somewhat equivalent to IT, but more encompassing in its scope.
2. A "bleeding edge" development is one in which new technology is being developed—the "blood is on the floor."

Acknowledgments

Thanks to Tony Dale and Robin Gauld, as well as the referees of this paper, for their assistance and suggestions. All errors are mine.

References

Abel-Hamid, Tarek K. 1988. Understanding the "90% Syndrome" in Software Project Management: A Simulation-Based Study. *Journal of Systems and Software* 8(4): 319–30.

Al-Gahtani, Said S., and Malcolm King. 1999. Attitudes, Satisfaction and Usage: Factors Contributing to Each in the Acceptance of Information Technology. *Behaviour and Information Technology* 18(4): 277–97.

Auditor General of Canada. 2002. *Report of the Auditor General of Canada to the House of Commons*. Toronto: Office of the Auditor General.

http://www.oag-bvg.gc.ca/domino/reports.nsf/html/02menu_e.html [accessed July 23, 2007].

Bascarini, D. 1999. The Logical Framework for Defining Project Success. *Project Management Journal* 30(4): 25–32.

Beetham, David. 1996. *Bureaucracy*. 2nd ed. Minneapolis: University of Minnesota Press.

Birnum, Robert. 2000. *Management Fads in Higher Education: Where They Come From, What They Do, Why They Fail*. San Francisco: Jossey-Bass.

Bovens, Mark, and Paul 't Hart. 1996. *Understanding Policy Fiascoes*. New Brunswick, NJ: Transaction.

Bovens, Mark, Paul, 't Hart, and B. Guy Peters, eds. 2001. *Success and Failure in Public Governance: A Comparative Analysis*. Cheltenham, UK: Edward Elgar.

Brindle, Margaret C., and Peter N. Stearns. 2001. *Facing Up to Management Faddism: A New Look at an Old Force*. Westport, CT: Quorum.

Bronson, Po. 1999. *The Nudist on the Late Shift and Other Tales of Silicon Valley*. London: New York: Random House.

Brown, Mary Maureen, and Jeffrey L. Brudney. 1998. A "Smarter, Better, Faster, and Cheaper" Government: Contracting and Geographic Information Systems. *Public Administration Review* 58(4): 335–45.

———. 2003. Learning Organizations in the Public Sector? A Study of Police Agencies Employing Information and Technology to Advance Knowledge. *Public Administration Review* 63(1): 30–43.

Bussen, Wendy, and Michael D. Myers. 1997. Executive Information System Failure: A New Zealand Case Study. *Journal of Information Technology* 12(2): 145–53.

Cohen, Michael D., James G. March, and Johan P. Olsen. 1972. A Garbage Can Theory of Organizational Choice. *Administrative Science Quarterly* 17(1): 1–25.

Cole, Andy. 1995. Runaway Projects: Cause and Effects. *Software World (UK)* 26(3): 3–5.

Collins, Tony, with David Bicknell. 1997. *Crash: Learning from the World's Worst Computer Disasters*. New York: Simon & Schuster.

Coltman, Tim, Timothy M. Devinney, Alopi Latukefu, and David F. Midgley. 2001. E-Business: Revolution, Evolution, or Hype? *California Management Review* 44(1): 57–86.

Connor, Patrick. 1997. Total Quality Management: A Selective Commentary on Its Human Dimensions, with Special Reference to Its Downside. *Public Administration Review* 57(6): 501–9.

Corner, Ian, and Matthew Hinton. 2002. Customer Relationship Management Systems: Implementation Risks and Relationship Dynamics. *Qualitative Market Research* 5(4): 239–51.

Dalcher, Darren, and Audley Genus. 2003. Avoiding IS/IT Implementation Failure. *Technology Analysis and Strategic Management* 15(4): 403–7.

Dale, Tony, and Shaun Goldfinch. 2002. Pessimism as an Information System Management Tool in the Public Sector: Lessons from the INCIS Fiasco in the New Zealand Police Force. http://nz.cosc.canterbury.ac.nz/research/reports/TechReps/2002/tr_0202.pdf [accessed July 23, 2007].

Dinsdale, Paul. 2004. Information Technology Is Not Meeting Nurses' Needs: Staff Want to Be Involved in the Development of the System They Are Expected to Use. *Nursing Standard* 19(7): 8–9.

Dizard, Wilson. P. 2001. CIOs Labor over Contracting Crisis. *Government Computer News*, November. http://www.gcn.com/state/7_11/news/16778-1.html [accessed July 23, 2007].

Du Gay, Paul. 2000. *In Praise of Bureaucracy: Weber, Organization, Ethics*. London: Sage Publications.

Dutton, William, Donald MacKenzie, Stuart Shapiro, and Malcom Peltu. 1995. Computer Power and Human Limits: Learning from IT and Telecommunications Disasters. PICT, Policy Research Paper No. 33.

The Economist. 2002. The Health Service's IT Problem. October 19, 51–52.

Evans, Mark, and Chris Reynolds. 2001. Software Barely Makes the Grade. *Computerworld New Zealand*, November 19, 43.

Fenton, Norman E., and Shari Lawrence Pfleeger. 1997. *Software Metrics: A Rigorous and Practical Approach*. 2nd ed. Boston: PWS.

Georgiadou, Elli. 2003. Software Process and Product Improvement: A Historical Perspective. *Cybernetics and Systems Analysis* 39(1): 125–42.

Ginzberg, Michael J. 1981. Key Recurrent Issues in the MIS Implementation Process. *MIS Quarterly* 5(2): 47–59.

Glass, Robert. 1998. Software Runaways—Some Surprising Findings. *DATABASE for Advances in Information Systems* 28(3): 16–19.

———. 1999. Evolving a New Theory of Project Success. *Communications of the ACM* 42(11): 17–19.

Grant, Gerald. 2003. Strategic Alignment and Enterprise Systems Implementation: The Case of Metalco. *Journal of Information Technology* 18(3): 159–75.

Grover, Varun, Albert L. Lederer, and Rajiv Sabherwal. 1988. Recognizing the Politics of MIS. *Information and Management* 14(3): 145–56.

Heeks, Richard, ed. 1999. *Reinventing Government in the Information Age: International Practice in IT-Enabled Public Sector Reform*. London: Routledge.

———. 2002. *Failure, Success and Improvisation of Information System Projects in Developing Countries.* Development Informatics Working Paper Series, No. 11/2002. Manchester, UK: Institute for Development Policy and Management.

———. 2004. *eGovernment as a Carrier of Context*. iGovernment Working Paper Series, No. 15. Manchester: Institute for Development Policy and Management, University of Manchester.

Heeks, Richard, and Subhash Bhatnagar. 1999. Understanding Success and Failure in Information Age Reform. In *Reinventing Government in the Information Age: International Practice in IT-Enabled Public Sector Reform*, edited by Richard Heeks, 49–74. London: Routledge.

Heeks, Richard, and Anne Davies. 1999. Different Approaches to Information Age Reform. In *Reinventing Government in the Information Age: International Practice in IT-Enabled Public Sector Reform*, edited by Richard Heeks, 22–48. London: Routledge.

Hewson, N., and W. Hewson. 1998. *ROI in Sales and Marketing Systems*. Cambridge, UK: Hewson Consulting Group.

Holden, Stephen. 2003. The Evolution of Information Technology Management at the Federal Level: Implications for Public Administration. In *Public Information Technology: Policy and Management Issues*, edited by G. David Garson, 53–73. Hershey, PA: Idea Group.

Huang, Jimmy, Enesi Makoju, Sue Newel, and Robert Galliers. 2003. Opportunities to Learn from "Failure" with Electronic Commerce: A Case Study of Electronic Banking. *Journal of Information Technology* 18(1): 17–26.

Iacovou, Charalambos L. 1999. The IPACS Project: When IT Hits the Fan. *Journal of Information Technology* 14(2): 267–75.

Irani, Zahir, A. M. Sharif, and Peter Love. 2001. Transforming Failure into Success through Organisational Learning: An Analysis of a Manufacturing Information System. *European Journal of Information Systems* 10(1): 55–66.

Jacobson, Ivar, James Rumbaugh, and Grady Booch. 1998. *The Unified Software Development Process*. Reading, MA: Addison-Wesley.

James, Geoffrey. 1997. IT Fiascoes and How to Avoid Them. *Datamation*, November, 84–88.

Keil, Mark, and Daniel Robey. 2001. Blowing the Whistle on Troubled Software Projects. *Communications of the ACM* 44(4): 87–93.

Keil, Mark, Joan Mann, and Arun Rai. 2000. Why Software Projects Escalate: An Empirical Analysis and Test of Four Theoretical Models. *MIS Quarterly* 24(4): 631–64.

Knights, David, and Fergus Murray. 1994. *Managers Divided: Organisation Politics and Information Technology Management*. Chichester, UK: Wiley.

Korac-Boisvert, N., and A. Kouzmin. 1995. Transcending Soft-Core IT Disasters in Public Sector Organizations. *Information Infrastructure and Policy* 4(2): 131–61.

Kraemer, Kenneth, and Jason Dedrick. 1994. Payoffs from Investment in Information Technology—Lessons from the Asia-Pacific Region. *World Development* 22(12): 1921–31.

Kwon, T. H., and R. W. Zmud. 1987. Unifying the Fragmented Models of Information Systems Implementation. In *Critical Issues in Information Systems Research*, edited by R. J. Bolland, Jr., and R. A. Hirscheim, 227–51. New York: Wiley.

Larsen, M., and M. Myers. 1999. When Success Turns to Failure: A Package-Driven Process Re-Engineering Project in the Financial Services Industry. *Journal of Strategic Information Systems* 8(4): 395–417.

Laudon, Kenneth C., and Jane P. Laudon. 1998. *Management Information Systems: New Approaches Organization and Technology*. Englewood Cliffs, NJ: Prentice Hall.

Lindblom, Charles E. 1959. The "Science" of Muddling Through. *Public Administration Review* 19(2): 79–88.

Lucas, Henry. 1981. *Implementation: The Key to Successful Information Systems*. New York: Columbia University Press.

Lyytinen, Kalle, and Daniel Robey. 1999. Learning Failure in Information Systems Development. *Information Systems Journal* 9(2): 85–101.

Mahaney, Robert, and Albert Lederer. 1999. Runaway Information System Projects and Escalating Commitment. *SIGCPR' 99*: 291–96.

Merton, Robert K. 1957. *Social Theory and Social Structure*. Glencoe, IL: Free Press.

March, James G., and Johan P. Olsen. 1996. Institutional Perspectives on Political Institutions. *Governance* 9(3): 247–64.

Nandhakumar, J. 1996a. Design for Success? Critical Success Factors in Executive Information Systems Development. *European Journal of Information Systems* 5(1): 62–72.

———. 1996b. Executive Information System Development: A Case Study of a Manufacturing Company. *Journal of Information Technology* 11(3): 199–209.

Norris, Donald F., and M. Jae Moon. 2005. Advancing E-Government at the Grassroots: Tortoise or Hare? *Public Administration Review* 65(1): 64–75.

Northrop, Alana. 2003. Information Technology and Public Administration: The View from the Profession. In *Public Information Technology: Policy and Management Issues*, edited by G. David Garson, 1–19. Hershey, PA: Idea Group.

Northrop, Alana, Kenneth Kraemer, Debora Dunkle, and John King. 1990. Payoffs from Computerization—Lessons over Time. *Public Administration Review* 50(5): 505–14.

Oakeshott, Michael. 1962. *Rationalism in Politics, and Other Essays*. London: Methuen.

Popper, Micha, and Raanan Lipshitz. 2000. Organizational Learning—Mechanisms, Culture, and Feasibility. *Management Learning* 31(2): 181–96.

Rocheleau, Bruce. 2003. Politics, Accountability, and Governmental Information Systems. In *Public Information Technology: Policy and Management Issues*, edited by G. David Garson, 20–52. Hershey, PA: Idea Group.

Royal Academy of Engineering, and British Computing Society. 2004. *The Challenges of Complex IT Projects*. London: Royal Academy of Engineering.

Salaman, Graeme. 2001. A Response to Snell: The Learning Organization: Fact or Fiction? *Human Relations* 54(3): 319–42.

SIMPL Group, and New Zealand Institute of Economic Research (NZIER). 2000. *Information Technology Projects: Performance of the New Zealand Public Sector in Performance*. Report to the Department of the Prime Minister and Cabinet. Wellington: SIMPL Group/New Zealand Institute of Economic Research.

Small, Francis. 2000. Ministerial Inquiry into INCIS. http://www.courts.govt.nz/pubs/reports/2000/incis_rpt/index.html [accessed July 24, 2007].

Smith, H. Jeff, Mark Keil, and Gordon Depledge. 2001. Keeping Mum as the Project Goes Under: Toward an Explanatory Model. *Journal of Management Information Systems* 18(2): 189–227.

Standish Group. 2001. Extreme Chaos. http://www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf.

———. 2004. Third Quarter Report 2004. http://www.standishgroup.com/sample_research/PDFpages/q3-spotlight.pdf.

Swanson, E. Burton. 1988. *Information System Implementation: Bridging the Gap between Design and Implementation*. Henley-on-Thames, UK: Alfred Waller.

Teega Associates. 2003. *Automation Program Evaluation*. Wellington, New Zealand: LINZ.

Tesser, A., and S. Rosen. 1972. Fear of Negative Evaluation and the Reluctance to Transmit Bad News. *Journal of Communication* 22(2): 124–41.

't Hart, Paul. 1994. *Groupthink in Government: A Study of Small Groups and Policy Failure*. Baltimore: Johns Hopkins University Press.

Turner, Ian. 1998. *Strategy, Complexity and Uncertainty*. http://www.poolonline.com/archive/iss1fea5.html [accessed July 24, 2007].

Wastell, David G. 1999. Learning Dysfunctions in Information Systems Development: Overcoming the Social Defenses with Transitional Objects. *MIS Quarterly* 23(4): 581–600.

Wilcocks, Leslie. 1994. Managing Information Systems in U.K. Public Administration: Issues and Prospects. *Public Administration* 72(1): 13–32.

Wilson, M., and D. Howcroft. 2002. Reconceptualising Failure: Social Shaping Meets IS Research. *European Journal of Information Systems* 11(4): 236–50.