December 2003

Including Technical and Security Risks in the Development of Information Systems: A Programmatic Risk Management Model

Robin Dillon
Georgetown University

INCLUDING TECHNICAL AND SECURITY RISKS IN THE DEVELOPMENT OF INFORMATION SYSTEMS: A PROGRAMMATIC RISK MANAGEMENT MODEL

Robin L. Dillon
McDonough School of Business
Georgetown University
Washington, DC USA
RLD9@georgetown.edu

Abstract

Developing and managing an information systems project has always been challenging, but with increased security concerns and tight budget resources, the risks are even greater. With more networks, mobility, and telecommuting, there is an increased need for an assessment of the technical and security risks. These risks if realized can have devastating impacts: interruptions of service, data theft or corruption, embezzlement and fraud, and compromised customer privacy. The software risk assessment literature (for example, Barki et al. 2001; Lyytinen et al. 1998; Schmidt et al. 2001) has focused primarily on managerial (i.e., development) risks, while the security risk models (for example, Cohen et al. 1998; Straub and Welke 1998) do not include the development risks and implementation costs. Theoretical risk models need to be developed that can provide a framework for assessing and managing the critical technical failure and security risk factors in conjunction with the managerial and development risks. This research seeks to model this problem by extending risk models originally developed for large-scale engineering systems.

Keywords: Information systems, reliability, security, probability, risk analysis, risk management

Research Objectives and Questions

Today's information systems projects are grasping with how to make the system open for the right users to access and share data but closed enough to keep the wrong users out. Risks must now include threats to the system (such as functionality and reliability) and threats to the data (such as integrity, confidentiality, and availability) (Denning 1999). However, the management risks (such as failure to gain user commitment and lack of frozen requirements) have not gone away. Also, in the current economic environment, the resources available for systems development are tightly constrained, thus requiring trade-offs between more risky, cutting-edge systems and more robust systems with modest functionality.

For almost four decades, research in information systems development and software risk assessment has cited statistics such as 46 percent of the development projects surveyed were completed over budget and past original deadlines and 28 percent were cancelled before completion (Standish Group 1998). In an attempt to remediate the continuing problem of information system failures, software engineering and information system researchers developed rigorous systems analysis and design methods (for example, Whitten and Bentley 1998) and conducted numerous surveys in an attempt to systematically organize critical risk factors (Barki et al. 2001; Schmidt et al. 2001). The systems development life cycle processes include steps for technical and operational feasibility studies but provide no models to accomplish these tasks. The critical risk lists provide valuable input to a risk management program to the degree that they help identify potentially difficult projects that require special attention or additional resources (McFarlan 1981). However, in the survey of risk factors compiled by Schmidt et al. (2001), not a single one of the risk factors had to do with any security aspects of the system, and Jiang and Klein's (2000) study of software development risks also does not include technical failure or security performance. In the information systems security literature, extensive lists of potential attacks, defenses, threats, and consequences, have been compiled (Cohen 1997a, 1997b; Cohen et al. 1998), but the models do not address management and implementation issues associated with the mitigation actions. While the performance and capabilities of both hardware and software have improved significantly over time, with more networks, mobility, and telecommuting, we need to assess and mitigate both technical failure and security risks in conjunction with management risk factors in the development and implementation phase.

The framework described here addresses this risk tradeoff problem including technical failure, security, and management risks, and is based on probabilistic risk analysis of the current information system. This probabilistic risk model in combination with decision analysis provides a decision support framework for resource allocation decisions during information systems development and for examining technical failure and operational feasibility as part of the life cycle design. The objective of this research is to demonstrate, for the development of an information system, how a project management framework based on a probabilistic model of the information system's performance, the risk factors, the risk mitigation options, and the design alternatives, can maximize the expected project outcome through the optimal allocation of project resources. The model uses a utility function to explicitly examine the tradeoffs between minimization of the probability of an IS project's failure (during development or operations) and maximization of the expected benefits from its performance.

The primary result of the research is a theoretical framework to guide design and resource allocation decisions to minimize the risks of information systems failures, both in development and in operations. This framework can be modeled using Excel and off-the-shelf decision and risk software to create a prototype decision support system to provide quantitative analysis of risk tradeoffs and resource allocations for information systems development projects.

Theoretical Foundations of the Study

The framework is based on probabilistic risk analysis (PRA) and decision analysis (DA) where PRA is used to quantify the risk of potential alternatives and DA provides the framework for including values and preferences to determine if the potential benefits are worth the associated risks. The PRA model (see, for example, Henley and Kumamoto 1992; Kaplan and Garrick 1981) links the reliability of individual components and the overall system configuration to quantify the overall technical failure risk. This approach to risk assessment is similar to that advocated by Boehm (1991) but requires the quantitative assessment of probabilities and outcomes. The primary objective of decision analysis is to determine which alternative course of action will maximize the expected utility for the decision maker. It is based on the existence of a set of logical axioms and a systematic procedure to aggregate probabilities and preferences based upon those axioms (Bodily 1992). Unique to decision analysis is the creation of a preference model to evaluate the alternatives and consequences (Keeney 1982).

The need for this decision-risk framework is justified by Barki et al. (2001), McFarlan (1981), and others in the software risk literature (for example, Jiang et al. 2001; Nidumolu 1996; Ropponen and Lyytinen 1997). Their research shows that for complex information system development problems, project management tools that help identify and mitigate risks are key factors in determining project success. Also, Keil et al. (1998) document the need to establish the relative importance of the risks so managerial attention can be focused on the areas that constitute the greatest threats, but their study included little discussion of technical or security risks. In the information security risk literature, Straub and Welke (1998) recommend a risk-decision framework to improve on crude cost-benefit mechanisms generally adopted.

Key structural components of the decision-risk framework as described further are derived from software risk management literature (for example, Barki et al. 2001; Nidumolu 1995) and also information security research (for example, Cohen 1997a, 1997b; Denning 1999; Greenstein and Feinman 2000).

Model Framework

As shown in Figure 1, managers must carefully balance information assurance and operational capability. For example, the more capabilities that you provide your employees to remotely access and alter files that they store on the network, the more security is required to prevent unauthorized users from accessing and altering network files. This balance must occur within the available budget resources and with consideration for all of the traditional management risks identified by the software development risk literature (for example, Barki et al. 1993; Boehm 1991; Schmidt et al. 2001) such as lack of top management commitment, misunderstanding the requirements, changing scope and objectives, etc. This distinction between management and technical

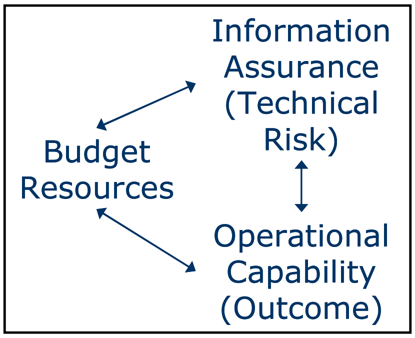
Information Assurance (Technical Risk)
Budget Resources
Operational Capability (Outcome)

Figure 1. Factor Trade-Offs

failure risk factors is consistent with the distinction made between process and product performance in the literature (Barki et al. 2001; Nidumolu 1995). Management risk factors identify potential problems during development (i.e., the process), and technical failure risk factors assess the product's likely success in operations. Security risks are those factors that specifically consider malicious attempts to access or compromise the system in some way. The optimal balance can be determined based on maximizing the expected utility for the design alternatives. The utility of the outcome is based on both the total costs spent (Z) and the operational capability of the final system (D) given that the system works. The system working is defined by a PRA model that quantifies the probability of technical/security failure (p(TF)), and the probability of a technical failure given investments in reinforcement can then be expressed as a function of the probabilities of the different failure modes based on the design configuration, the investments in the reinforcement of the components, and the effects of these investments on the component reliability.[1] Budget resources initially held in reserve (R) are required to mitigate development problems that occur. Resources not held in reserve can be spent to enhance the operational capability of the system (E) and to reinforce the technical reliability/information assurance (I). The total costs spent (Z) includes all resource categories, and the more that is committed up-front to I and E (i.e., not held in reserve), the greater the likelihood for cost overruns if development/management problems are realized. The expected utility of an alternative (A) is thus:

$$EU(A) = U\left(Z, D_A \middle| E\right) \times \left(1 - p\left(TF \middle| I\right)\right) \qquad (1)$$

The decision maker thus faces two types of uncertainty: (1) the possibility of development problems (e.g., specific functions may not be completed on time, etc. (see Barki et al. 2001; Boehm 1991; Schmidt et al. 1991) and (2) the system's performance in operations such as security breaches (Cohen 1997a). The optimal design and the level of reserves are then chosen to maximize the decision maker's overall utility function for the system based on these factors. Figure 2 provides a graphical representation of the interaction of the variables in the model where the rectangles represent decisions, the circles are uncertainties, the rounded rectangles are outcomes, and the diamond is the overall value or expected utility. This framework was originally developed based on case studies of space projects (Dillon et al. 2003) and has been modified here to include information technology specific-risks and factors.

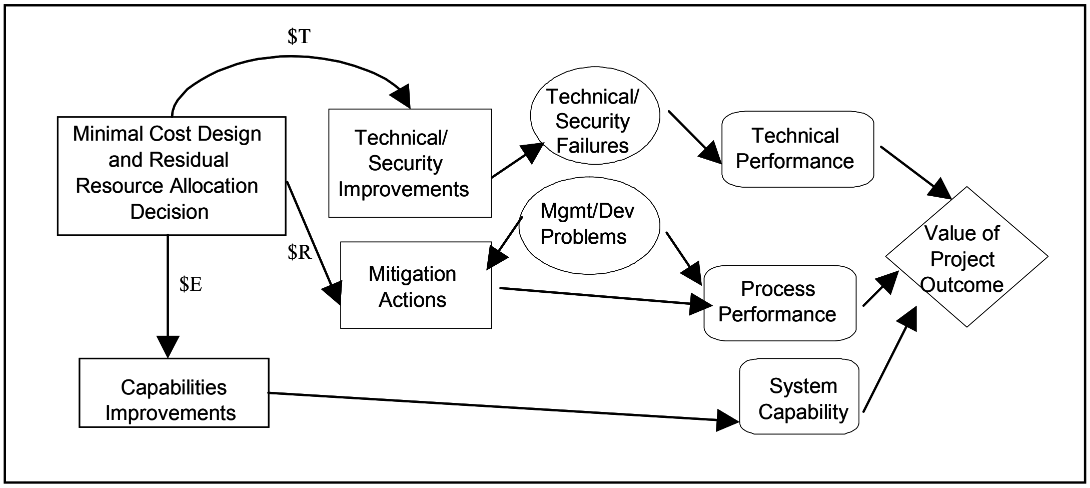
$T
Minimal Cost Design and Residual Resource Allocation Decision
Technical/ Security Improvements
Technical/ Security Failures
Technical Performance
Mgmt/Dev Problems
$R
Mitigation Actions
$E
Process Performance
Value of Project Outcome
Capabilities Improvements
System Capability

Figure 2. Influence Diagram Showing Relationship of Model Variables

[1] Measures for each model variable are developed as the first step in the case studies proposed in the next section and will be discussed in the presentation.

Consider briefly a Web server example. Functions include verifying accounts, storing files, serving requested data, tracking users and creating logs, providing maintenance and administrative capabilities, and ensuring security. Example failures include the system is not available and/or the user cannot access it, data confidentiality is lost, or the integrity of the server is lost either from proper data being corrupted or from improper data or files being added. These failures can result from several initiating events as shown in Figure 3 including attempted attacks, system administrator errors, hardware or software failures, or some transient events (e.g., cut cables or power outages). In Figure 3 (as in Figure 2), circles are uncertainties and the diamond is the overall value, in this case, the total cost consequences of failure. Cost consequences may include losses from fraud, lost revenues from lower sales or fewer users, costs from lost time in terms of productivity, and/or the intrinsic value of the lost data. Example development alternatives include firewalls (various hardware and software alternatives), different operating system alternatives, various hardware configurations of multiple servers, encryption tools, and improved development and maintenance processes.

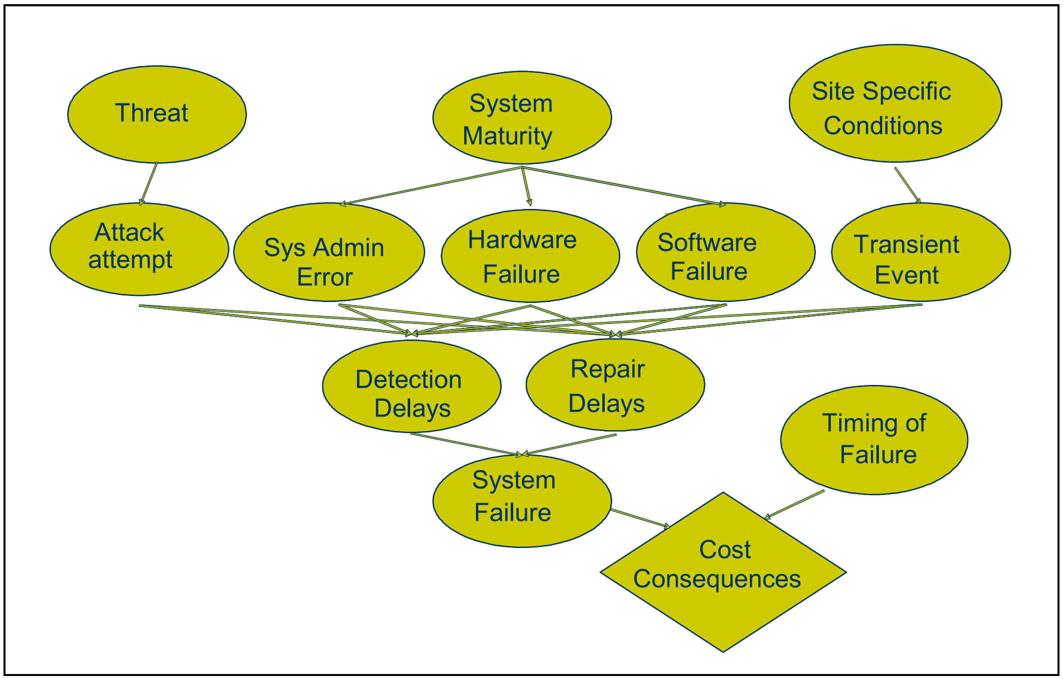

Figure 3. Influence Diagram Showing the Initiating Events and Risks to System Failure

Assume that the organization estimates that the relative costs of security losses are as follows: 17 percent of losses from viruses (in terms of productivity), 71 percent of losses from unauthorized internal logical (rather than physical) access from fraud, and 12 percent for all other types.[2] The magnitude of losses is estimated between $1.1 billion and $2.4 billion (assume a uniform distribution). In order to reduce fraud costs, the organization is planning on implementing a Public Key Infrastructure (PKI). The PKI is a combination of software, hardware, and procedures that provide secure and confidential data transfer using cryptography. The costs for an organization-wide roll-out are about $1.8 million annually. Implementing PKI organization-wide is expected to reduce fraud incidents by 0.11 percent. Also, assume that users will be inconvenienced in 1 out of every 10,000 transactions from failures of certificating authorities, expired keys, or corrupted keys (assume 100,000 transactions per day and an inconvenience cost of $100 per incident). In rolling out the PKI, the organization has two alternatives: (1) organization-wide or (2) targeted. In a targeted roll-out, it is estimated that providing PKI to a key fraction of the users (50 percent) costs 65 percent of the total cost, and will achieve 60 percent of the fraud avoidance benefits. Should PKI be implemented organize-wide or targeted?

[2] This example is loosely based on some analysis work performed to evaluate an information security program at the Department of Veterans Affairs. The work is documented in *Information Technology Performance Management: Measuring IT's Contribution to Mission Results*, IT Performance Management Subcommittee, Federal CIO Council, September 2001. The author *did not* participate in the original analysis.

Investing in the system organization-wide will reduce the probability of a technical failure by 0.11 percent, i.e., an expected loss reduction of $1.925 million. For the decision maker's utility, we assume a linear function measurable in dollars. Organization-wide inconveniences based on the assumed data are $365,000 per year. Thus for the organization-wide alternative, the expected costs (implementation plus inconvenience) exceed the expected benefits by $240,000 and for the limited implementation, the corresponding value is $197,500. Therefore, based on the technical failure and security risks and the benefits and assuming an expected value decision maker, the best alternative is the limited implementation. To complete the model, the next step is to consider how realistic the $1.8 million budget is (i.e., probability of overruns), what constitutes a budget overrun, and what other alternatives are available for that budget.

Current State of the Project and Conference Presentation

This example is a simple illustration of the types of data needed and the approach described by the proposed framework. The primary benefit is that it provides a proactive approach to resource management and risk identification. The framework forces the decision maker to consider security issues in development, rather than after the fact as is generally done in reality. This is important because many decisions made in development, such as the choice of the operating system for a Web server, have an impact on security. The research will continue with a series of case studies (Klein and Myers 1999) applying the framework to actual information systems development decisions. We have chosen a case study approach rather than a survey because of the biases identified by previous research regarding managers' perceptions of risk (Goodhue and Straub 1991, Schmidt et al. 2001). In the conference presentation, we will explain the results of these case studies. An interesting outcome that will be discussed is the impact of management factors on the system's technical/security performance.

References

Barki, H., Rivard, S., and Talbot, J. "An Integrative Contingency Model of Software Project Risk Management," *Journal of Management Information Systems* (17:4), Spring 2001, pp. 37-69.

Barki, H., Rivard, S., and Talbot, J. "Toward an Assessment of Software Development Risk," *Journal of Management Information Systems* (10), 1993, pp. 203-223.

Bodily, S. "Introduction: The Practice of Decision and Risk Analysis," *Interfaces* (22:6), November/December 1992, pp. 1-4.

Boehm, B. W. "Software Risk Management: Principles and Practices," *IEEE Software*, January 1991, pp. 32-41.

Cohen, F. "Information System Attacks: A Preliminary Classification Scheme," *Computers & Security* (16), 1997a, pp. 29-46.

Cohen, F. "Information Systems Defences: A Preliminary Classification Scheme," *Computers & Security* (16), 1997b, pp. 94-114.

Cohen, F., Phillips, C., Swiler, L. P., Gaylor, T., Leary, P., Rupley, F., and Isler, R. "A Cause and Effect Model of Attacks on Information Systems," *Computers & Security* (17), 1998, pp. 211-221.

Denning, D. *Information Warfare and Security*, Addison-Wesley, Boston, 1999.

Dillon, R. L., Paté-Cornell, M. E., and Guikema, S. "Programmatic Risk Analysis for Critical Engineering Systems under Tight Resource Constraints" *Operations Research* (51:3), May-June 2003, pp. 354-370.

Goodhue, D., and Straub, D. "Security Concerns of System Users: A Study of Perceptions of the Adequacy of Security Measures," *Information and Management* (20:1), January 1991, pp. 13-27.

Greenstein, M., and Feinman, T. M. *Electronic Commerce: Security, Risk Management, and Control*, Irwin McGraw-Hill, Boston, 2000.

Henley, E., and Kumamoto, H. *Probabilistic Risk Assessment: Reliability Engineering, Design, and Analysis*, IEEE Press, New York, 1992.

Jiang, J., and Klein, G. "Software Development Risks to Project Effectiveness," *The Journal of Systems and Software* (52), 2000, pp. 3-10.

Jiang, J. J., Klein, G., and Discenza, R. "Information System Success as Impacted by Risks and Development Strategies," *IEEE Transactions on Engineering Management* (48), 2001, pp. 46-55.

Kaplan, S., and Garrick, B. J. "On the Quantitative Definition of Risk," *Risk Analysis* (1:1), 1981, pp. 11-27.

Keeney, R. "Decision Analysis: An Overview," *Operations Research* (30:5), September/October, 1982, pp. 803-838.

Keil, M., Cule, P., Lyytinen, K., and Schmidt, R. "A Framework for Identifying Software Project Risks," *Communications of the ACM* (41:11), November 1998, pp. 76-83.

Klein, H. K., and Myers, M. D. "A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems," *MIS Quarterly* (23:1), March 1999, pp. 67-89.

Lyytinen, K., Mathiassen, L., and Ropponen, J. "Attention Shaping and Software Risk A Categorical Analysis of Four Classical Risk Management Approaches," *Information Systems Research* (9:3), September 1998, pp. 233-255.

McFarlan, F. W. "Portfolio Approach to Information Systems," *Harvard Business Review*, September/October 1981, pp. 142-150.

Nidumolu, S. R. "A Comparison of the Structural Contingency and Risk-Based Perspectives on Coordination in Software Development Projects," *Journal of Management Information Systems* (13:2), Fall 1996, pp. 77-113.

Nidumolu, S. R. "The Effect of Coordination and Uncertainty on Software Project Performance: Residual Performance Risk as an Intervening Variable," *Information Systems Research* (6:3), September 1995, pp. 191-219.

Ropponen, J., and Lyytinen, K. "Can Software Risk Management Improve System Development: An Exploratory Study," *European Journal of Information Systems* (6), 1997, pp. 41-50.

Schmidt, R., Lyytinen, K., Keil, M., and Cule, P. "Identifying Software Project Risks: An International Delphi Study," *Journal of Management Information Systems* (17:4), Spring 2001, pp. 5-36.

Straub, D., and Welke, R. "Coping with Systems Risk: Security Models for Management Decision Making," *MIS Quarterly*, December 1998, pp. 441-469.

The Standish Group, *1998 Chaos Report*, Dennis, MA. 1998.

Whitten, J. L., and Bentley, L. D. *Systems Analysis and Design Methods* (4th ed.), Irwin McGraw-Hill, Boston, 1998.