



Niv Ahituv, Moshe Zviran, and Chanan Glezer

# Top Management Toolbox for Managing Corporate IT

***These checks and balances prevent project runaways and deliver systems that further corporate strategy and performance.***

**T**he rapid introduction of new IT into organizations and the increased competition in the general business environment make managing the corporate IS function increasingly difficult. Yet applications developed by IS play an increasingly strategic role in the business performance of the entire organization [12].

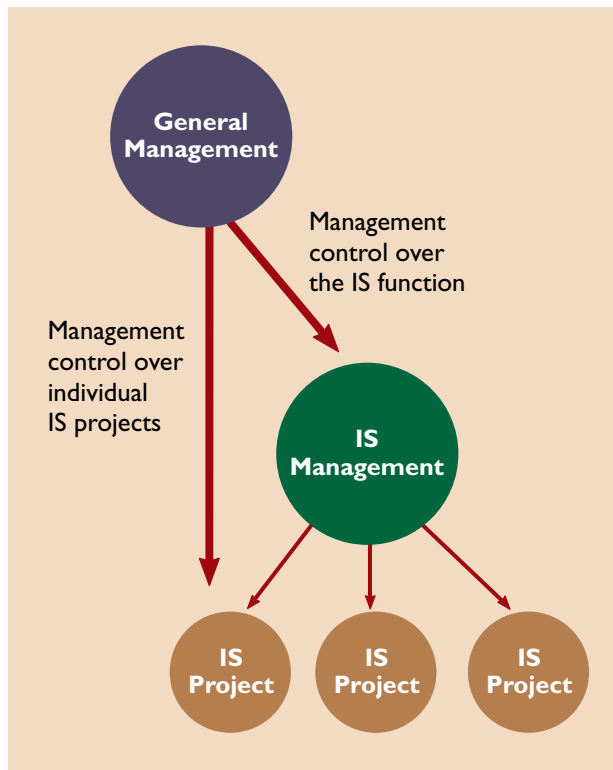
Developing strategic applications is a complex process involving such activities as identifying promising application targets, deciding the application's posture (defensive or offensive) and mode of use (internal or external), and assessing the application's associated risk of system failure and financial loss. One practical conclusion for strategic systems is that planning for them is intertwined with an organization's business strategy, so cooperation between IS managers and non-IS managers is critical for their implementation.

In many organizations, general and functional managers—with law, finance, accounting, marketing, or even engineering backgrounds—are not familiar with the latest technology. This raises a barrier to communication that limits their ability to determine risk before deciding whether to go ahead with a project or how to control large projects during their development life cycles.

Two examples illustrate how it can lead to disastrous results:

**Bank of America.** Development of a system to support the bank's management of trust accounts

was budgeted at \$25 million and scheduled to take two years to complete. The result was to be the MasterNet system, with online updating and automated generation of monthly statements. Instead, completion took five years at a total cost of \$80 million—and was still rejected by its target users because it created accounting problems and suffered from slow response times and communication and disk drive problems. Many of the bank's employees were laid off during development, and the bank's reputation was severely damaged. Some major banking customers even lost confidence in the bank; the total number of institutional accounts dropped from 800 to 700, and assets under management shrank from \$38 billion to \$34 billion [1, 2]. In 1995, seven years after the project's failure, the bank's management sold its institutional trusts and securities services division, which still needed a major technology investment to position itself to compete against the enormous institutions that dominate the field of institutional trust banking. However, because of the bank's flawed history in implementing technology to support corporate trust businesses, the bank's executives decided it



**Figure 1.** Channels of corporate control over IS activities.

would be better to unload the division and spend the money on more promising ventures.

**AMR.** The goal was development of the CONFIRM system, a comprehensive travel industry reservation system combining airline, car rental, and hotel reservation information. In 1988, a consortium including Hilton Hotels, Marriott Corp., and Budget Rent-A-Car Corp. subcontracted this large-scale project to AMRIS, a systems consulting firm operating as a subsidiary of American Airlines Corp.

The project suffered delays caused by malfunctions and was finally scrapped three and a half years after it began at a total loss of \$125 million. A post-mortem analysis found that the consortium participants had been misled by the CONFIRM project managers, who failed to report problems in the project's database, decision support, and integration technologies. Another cause of the failure was the infrequency (only once a month) of meetings among the consortium participants and the developers [9]. Nevertheless, the consortium participants continued investing in the project, even after the problems came to light.

The epidemic of project runaways, that is, projects in which cost and time far exceed what was planned, is also affected by the emerging issue of connectivity in communication systems. Twenty

years ago, most projects involved self-contained units and had minimal interfaces with other units. Today, because of the need to integrate information from many sources, as in data warehouse applications, the probability of technical and operational problems that cannot be foreseen at the beginning of a project has increased dramatically. To avoid such runaways, we recommend the following management guidelines [11]:

- Involve the user at an early stage;
- Put senior, nontechnical management in charge of the project to ensure it is completed on time and within budget;
- Set up milestones as interim deadlines;
- Insist on performance clauses holding suppliers legally responsible for meeting deadlines; and
- Do not update the requirements before the original plan is finished.

Software project management should seek to structure management of the IS function and its projects to help prevent such runaways. Work in this area focuses on pre-implementation risk analysis for individual projects, as well as for the overall project portfolio [2, 8]. Another project-management area, which has been researched extensively, involves software process models dealing with individual software projects, providing the order of stages and the criteria for transition between them [3].

These approaches and others have focused on assessing the risk of individual projects or of a portfolio of projects as a way to exercise control. Here we focus on the link between general management and the IS function in the interests of continuous control over the IS function. We endorse a checks-and-balances approach that can be applied by top management through a set of tools (the "toolbox") to help monitor critical projects (the micro level) as well as the overall functioning of organizational IT (the macro level).

Control is a process for ensuring that employees and subcontractors of an organization, in any of its functions and levels, act in ways that lead to achievement of organizational goals. Control is a critical aspect of effective managerial practices, implying objective standards of judgment. The first step is to establish performance standards, the next is to measure current performance, and the last is to take corrective action when there is a problem [5, 10].

The purpose of control in the context of the IS function is to ensure the technology progresses in line with management's policy and overall business strategy. Management wants technology exploited to serve

organizational goals, rather than to let the organization be pushed or compelled by the technology. This support is achieved most readily by strengthening the link between general management and IS-related activities and by monitoring critical projects. Control reduces the chances of critical-project runaways by ensuring that general management stays informed about a project's status throughout its life cycle, rather than being told when the project is about to fail—and can no longer be rescued. Figure 1 shows both possible channels of control from the perspective of general management.

The first channel deals with general management “bypassing” IS function management, getting directly involved in strategic or critical projects requiring close supervision. These projects usually represent a major expense and high technical and financial risk or introduce a new core technology to the organization (such as innovative prototyping or pilot testing). The second channel deals with general management supervising the IS function without getting into the details of daily operation or specific projects.

Several frameworks have been proposed for implementing control over individual IS projects. The most popular are the cost-and-fix model, the stagewise model, the waterfall model, and the evolutionary model [3]. However, most of them are geared toward control of individual projects by IS professionals, not by general business management. A risk-driven approach, the “spiral model,” was described in [4]; it presents the project as a cyclical rather than a linear process, generating the final system through several prototype iterations. This approach can accommodate all the other models as special cases and provide guidance as to which combination of models best fits a given software situation.

In an effort to control the IS function, the portfolio approach to IT development described in [8] starts by categorizing the inherent risk of individual projects through a questionnaire that addresses three project

| Dimension                         | Description                                                                                                                                                |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Project size</i>               | The absolute and relative monetary expense is positively correlated with risk in an IS project.                                                            |
| <i>Experience with technology</i> | The organization's familiarity with application languages, databases, hardware, and operating systems is negatively correlated with risk in an IS project. |
| <i>Project Structuredness</i>     | The nature of the task, or degree of structure of the expected outputs, are negatively correlated with risk in an IS project.                              |

**Table 1.** Dimensions of risk in an IS project portfolio.

| Name                                  | Locus of Control      | Method of Control |
|---------------------------------------|-----------------------|-------------------|
| Code and fix model                    | Individual project    | Process model     |
| Stagewise model                       | Individual project    | Process model     |
| Waterfall model                       | Individual project    | Process model     |
| Evolution model                       | Individual project    | Process model     |
| Transform model                       | Individual project    | Process model     |
| Spiral model                          | Individual project    | Process model     |
| Project risk assessment questionnaire | Individual project    | Risk assessment   |
| IT portfolio approach                 | Project and portfolio | Risk assessment   |

**Table 2.** Methods of control.

dimensions—size, organizational experience with the technology, and degree of structure (see Table 1)—and serves as a control mechanism for the project.

Table 2 lists the current methods for achieving control over the IS function or over individual projects. The common denominator among them is a focus on preventing project runaways. Early process models, including code and fix, stagewise, waterfall, evolutionary, and transform, approach this goal by proposing a recipe for the software development process. The spiral model and the model underlying the risk assessment questionnaire [2, 3] take different approaches, basing their frameworks on prediction of the potential risk in a project and identification of risky projects. The portfolio approach expands this scope to a portfolio of several IS projects rather than focusing on a single project. If it adopts the portfolio approach, the organization has to ensure a good fit between the risk associated with the portfolio and the relevant environmental characteristics, such as business competition and the technical proficiency of its IS personnel.

Nevertheless, none of these approaches emphasizes a continuous link between general management and the IS function or an individual project as a way

to control the risk of IS activities. We emphasize an approach and a set of control tools to fill this gap and provide continuous monitoring capabilities for general (non-IS) managers over the IS function and critical IS projects within an organization, based on the principle of checks and balances.

### How Much Control?

The checks-and-balances approach helps establish management control over the IS function (in general) and over strategic IT projects (in particular). It requires that for each authority within the IS function another authority reside inside or outside IS to check its products and provide equilibrium in terms of professional abilities and organizational influence. This approach is valid for providing internal balance within the IS function and for helping corporate management control the IS function as a whole.

Equilibrium within the IS function can be achieved in a number of ways. For example, the development function is balanced by an implementation function. The former is responsible for analyzing and coding the application, the latter for training users. The implementation function also provides feedback for the development function, serving to improve the system. The development function is also balanced by a methodological expert. The former represents the creative effort of the IS function, the latter ensures that creation of the new software meets its intended work standards (such as proper documentation). The operations function is balanced by a technical support group. The former is responsible for running the hardware, communication network, operating systems, help desk, and application; the latter includes experienced employees who verify that the system and the communication network support the actual targeted workload. And the development and operations functions are balanced by a data security function that protects the information and prevents damage to the system.

The desired balance between the entire IS function and the other functions inside and outside the organization can be achieved by weighing the following factors:

**End users.** The IS function incorporates requirements of both managerial and operational users into its development and maintenance efforts. An example of the need for such balance is a U.S. factory that introduced a software package for manufacturing control. The package required production line employees to enter three computer commands before initiating each production activity—and to enter even more commands after the activity was

completed. The users eventually abandoned the system, claiming it disrupted their work; as a result, the company had to replace the system with one that was much easier to use.

**Staff function.** In large organizations, a staff officer is usually assigned to represent general management vis-a-vis the IS function. This person has to be computer literate and have a wide organizational perspective yet cannot be a subordinate of the director of the IS function. The main areas of this person's activity are IS planning, resource allocation, and evaluation of the products vs. the plans.

**Human engineering.** This function is responsible for the fit between IS and the organization's work procedures and processes. For example, the terminology used by the software should reflect the terminology used by the employees, and adequate help features should be provided by the system's interface, which should be friendly and comply with the end users' work environment.

The checks-and-balances approach adds overhead in terms of employees and work tasks, but its long-term benefits are clear. In small and midsize organizations, some of these functions can be merged. For example, technical support, data security, and methodological control can be performed by the same employees, and system developers can train end users. But even in small organizations, it is desirable to maintain at least some separation between the system developer, the person defining the initial requirements, and the person who will use the final product.

### Management Toolbox

The checks-and-balances approach is not just a theoretical principle. Through a set of managerial and financial tools, it is designed to assist general management in monitoring the IS function and its strategic projects. The tools complement IS organizational balances and are applied at two levels:

- Macro. Control by general management over the IS function without getting into the details of individual projects; and
- Micro. Direct control by general management over critical IS projects within the IS function (such as introducing new workstations and developing strategic applications), whereby general management becomes a partner with the project manager and IS function managers in navigating these critical projects.

These tools can be classified into two categories, depending on their instrumentation: managerial,

including procedures, reports, and human “sensors,” and financial, based mainly on monetary and budgetary control. The toolbox can be divided into four drawers, constituting a 2D grid (see Table 3). For example, macro/financial tools include:

**Budgeting.** Budgets specify the resources committed to an annual plan and are usually stated in monetary terms or in terms of units of products and labor. They provide a powerful means of control, because they serve as a standard against which actual performance is compared. For the IS function, the budget is divided into various relevant items, including maintenance, development, hardware, software, training, staffing, and project outsourcing.

**Periodic budget monitoring.** Since budgets and timetables are based on estimates, some deviation is permissible, even natural. Expenditures that significantly exceed budget guidelines indicate a problem with either the original estimates or the quality of management. Such situations call for intervention by a steering committee, which can reexamine the budget, replace personnel, or even call off or postpone whole projects.

**User chargeback mechanism.** This mechanism can help involve users in projects, letting them express their priorities. Chargeback can be assigned to development activities, operations activities, or to both.

Macro/managerial tools include:

**Steering committee.** The members of this high-level policy-making unit are top managers representing the organization’s major functions, including IS. The steering committee mixes business expertise and IS know-how to focus on IS-related issues at the highest organizational levels. The committee is involved in long-range IS planning, setting project priorities, resource allocation, progress tracking, and conflict resolution. Participation by IS and non-IS managers in the same committee ensures a better fit between the IS and corporate strategies and reduces the planning burden of top IS managers. The committee should be chaired by the organization’s CEO.

**IS auditor.** This person reports to corporate management and specializes in evaluating IS controls as part of the organization’s general audit func-

| Level of Control | Locus of Control | Economic Tools                                                                                                  | Managerial Tools                                                                                                                                                 |
|------------------|------------------|-----------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Macro            | IS Function      | Budgeting<br>Periodic budget monitoring<br>User chargeback mechanism                                            | Steering committee<br>DP auditor<br>Long-range/annual planning                                                                                                   |
| Micro            | Specific Project | Continuous cost/benefit analysis<br>Project budgeting<br>Budget deviation analysis<br>User chargeback mechanism | Feasibility analysis<br>Schedule monitoring<br>Steering committee<br>SDLC standards<br>Quality assurance<br>Project management and CASE tools<br>Project manager |

**Table 3.** Toolbox for IS management control.

tion. The auditor studies the whole IS function, interviewing key users and IS personnel. Based on the interviews, document reviews, and validation of financial statements, the auditor produces a summary status report providing an objective assessment of the IS function.

**Long-range planning.** Looking at a horizon of three to five years, long-range planning (the master plan) generates a description of the IS function’s general projected direction, along with the resources (such as human resources, facilities, software, and hardware) needed to carry out its strategies. The long-range plan is a cooperative task of general management and IS management and generally helps reduce the communication barrier between them, since it conveys the priorities and preferences of top management to IS management.

**Annual planning.** This tool derives from a short-range plan following the master plan. The short-range plan addresses the application development and maintenance plans, including objectives, resources, schedules, and key project dates. The short-range plan describes the preferences of general management regarding priorities and resource allocation for different projects running over the next year or two.

Micro/financial tools include:

**Continuous cost/benefit analysis.** Most investments, including those in IS projects, have to be financially justifiable. Cost-benefit analysis over the course of a project helps general management deter-

mine whether the benefits it produces outweigh its cost. Using any evaluation approach, the value of the project at a given point in time is calculated by subtracting the value of the future flow of expenses from the value of the future flow of benefits. The ratio of benefits and expenses can be compared to a threshold at which the project is considered successful. The assessment should be repeated periodically to accommodate unexpected changes. Intangible benefits should also be reassessed periodically.

**Budgeting.** The project manager has to prepare a detailed project budget, including expenses for such components as human resources, hardware, and software at various stages. Several techniques help estimate project cost. For example, if another system was developed in the same environment, an analogy to such a system can be used to create the budget. In other cases, the expected total number of lines of code in the project or the quality of the product and the characteristics of the development environment can be used to plan the budget (as in the COCOMO model [3]).

**Budget deviation analysis.** General management views the project manager as responsible for deviations from a project's budget. By comparing the actual performance of the project team with the budget, general management and the project manager can identify symptoms that could result in a runaway situation. The analysis should be done frequently, because the sooner a problem is detected, the lower will be the cost to correct it.

**User chargeback mechanism.** Users are charged for activities during a system's development, testing, and operation, enabling the project manager to track expenses and involve users in decisions regarding the development process.

Micro/managerial tools include:

**Feasibility analysis.** A feasibility analysis establishes whether a project should be carried out and how it should be done. It should be performed collaboratively by a user representative and an information analyst and reviewed by general management. This analysis has to determine whether it is feasible to develop a system—from organizational, financial, and technological perspectives. It suggests several approaches for addressing the current problem, recommends one of them, and outlines a project charter. The involvement of general management in the review process facilitates management control over IS projects from their earliest phases.

**Schedule monitoring.** A project is organized into activities performed in a predetermined sequence. Activities can be classified as critical and noncritical. A critical activity must be completed on schedule. A

noncritical activity has some slack time during which it may be delayed. The project manager and general management are interested in the bottlenecks created whenever a critical activity is delayed, thereby delaying the whole project. Thus, these activities are the first and most important to be monitored. However, attention should also be given to noncritical activities to avoid delays beyond the permitted slack.

**Project steering committee.** In addition to the IS steering committee, an ad hoc project steering committee should be formed to monitor a project's timetable and budget and ascertain that the final product—the system—fulfills the original requirement that prompted the organization to undertake it in the first place. The committee includes senior managers representing users and general management, as well as the director of the IS function, or CIO. It is recommended that the committee meet once a month to review the project's progress and adjust the original plan as needed. The committee deals with a wide range of topics related to the project's success, such as functional criteria, key appointments, cost-benefit analyses, acquisitions, prototype testing and evaluation, and transitions from one life cycle stage to the next.

**Life cycle.** A system's life cycle can be organized into several phases, including definition, construction, implementation, and operation. Each phase is further organized into a set of activities. For each activity, the appropriate people are assigned to help prepare documentation and authorize transition to the next activity. The projected life cycle serves as a checklist enabling general management to trace a project's progress at any given time and receive written documentation, thus reducing the chances of a project runaway [7].

**Quality assurance.** The project's management process and the system being developed should be evaluated by general management and the project steering committee in terms of quality measurements. Regarding the project management process, general management wants to verify compliance with the framework of the life cycle stages, budgets, and timetables. Other important control tools include software engineering methodologies (such as configuration control, structured analysis, data and process dictionaries, and data modeling), frankness of reporting, and project management documentation.

Regarding the system as a product of all these steps, methods, and tools, general management wants to know if it is flexible, easy to maintain, compatible with the organization's existing and future systems, efficient, and expandable. Software docu-

mentation serves as a major vehicle for communicating such information to general management.

**Project management and computer-aided software engineering (CASE).** These tools help schedule, control, and communicate about project activities. Project managers normally use a computerized package that formally describes the project and optimizes activities and resources during its development process (such as Gantt, PERT/CPM). CASE tools help software developers plan, analyze, design, program, and maintain systems by creating computerized user requirement specifications, dataflow diagrams, data models, a data dictionary, and structure charts. They also add an extra layer of organization to the project, making it more understandable by general management.

**Liaison officer.** This person serves as a liaison between users and the various functions in the IS department, coordinating all project-related activities, including analysis, design, programming, and testing, while communicating requests from end users to IS and reporting project status to end users. Managers improve their control over the project, because the liaison officer delivers a clear and reliable picture of the project.

As a general rule, whenever a project has a higher level of risk—due to size, innovation, or complexity—we recommend using more than one tool for greater control. In many organizations, a system is developed gradually, evolving out of the current system as a result of operational problems or requests for changes from end users. Control tools should be used in such cases, not only for projects starting from scratch.

## Conclusions

The need for better management control and involvement in IS activities stems from two motivations—runaway IS projects and the rapid growth in the use of IS to advance overall corporate strategy. Behind them is the deep problem of inadequate control and a communication barrier between nontechnical managers and IS managers. The checks-and-balances approach provides an innovative dimension for managing IS activities in organizations, requiring that for each authority within the IS function there is another authority inside or outside IS checking its products and providing a balance between IS's professional abilities and overall organizational influence.

The checks-and-balances approach is valid for providing internal balance within the IS function and for general management's interests in controlling the overall IS function. So instead of using a risk-based method for controlling individual pro-

jects or assessing a portfolio of risk through a snapshot questionnaire with uncertain validity, the checks-and-balances approach and its corresponding toolbox function as a comprehensive safeguard for minimizing risk caused by the communication barrier between IS managers and general managers.

However, the checks-and-balances approach cannot guarantee a risk-free environment but does provide general management an early warning about problems in the IS function by ensuring an appropriate balance between IS and the rest of the organization and within the IS function itself.

This approach should be used as a guideline in the design, construction, operation, and modification of an organization's technology functions. Each activity or function should reflect a balance (internal or external) utilizing at least one of the control tools. Determining the appropriate balance and which tools to use should be based on an organization's experience with similar projects, on the history of other organizations with similar corporate profiles, and on the managerial style in each specific organization. ■

## REFERENCES

1. Ahituv, N., Neumann, S., and Riley, H. *Principles of Information Systems for Management*. 4th ed. Brown Communications, Inc., Dubuque, Iowa, 1994.
2. Barki, H., Rivard, S., and Talbot, J. Towards assessment of software development risk. *J. Manage. Inf. Syst.* 10, 2 (Fall 1993), 203–225.
3. Boehm, B. *Software Engineering Economics*. Prentice-Hall, Englewood Cliffs, N.J., 1981.
4. Boehm, B. A spiral model of software development and enhancement. *Comput.* 21, 5 (May 1988), 61–72.
5. DuBrin, A., Ireland, R., and Williams, J. *Organization and Management*. South-Western Publishing Co., Cincinnati, Ohio, 1989.
6. Frantz, D. \$80-million failure: B of A's plans for computer don't add up. *Los Angeles Times*, Feb. 7, 1988, D1–D3.
7. McCracken, D., and Jackson, M. Life-cycle concepts considered harmful. *ACM Software Eng. Notes* (Apr. 1982), 29–32.
8. McFarlan, F. Portfolio approach to information systems. *Harvard Bus. Rev.* 59, (Sept.–Oct. 1981), 17–25.
9. Oza, E. When professional standards are lax: The CONFIRM failure and its lessons. *Commun. ACM* 37, 10 (Oct. 1994), 29–36.
10. Robey, D., and Sales, C. *Designing Organizations*. 4th ed. Irwin, Chicago, Ill., 1994.
11. Rothfeder, J. It's late, costly, incompetent, but try firing a computer system. *BusinessWeek* (Nov. 7, 1988), 86–89.
12. Zwass, V. *Management Information Systems*. Brown Communications, Inc., Dubuque, Iowa, 1992.

---

NIV AHITUV (ahituv@post.tau.ac.il) is the Lucie and Marko Chaoul Professor of Research in Information Evaluation on the Faculty of Management at Tel Aviv University in Tel Aviv, Israel. MOSHE ZVIRAN (zviran@post.tau.ac.il) is a senior lecturer on the Faculty of Management at Tel Aviv University in Tel Aviv, Israel. CHANAN GLEZER is a research associate on the Faculty of Management at Tel Aviv University in Tel Aviv, Israel.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

---

© 1999 ACM 0002-0782/99/0400 \$5.00