Risk Management for Laboratory Automation Projects

R. D. McDowall
McDowall Consulting, Kent, United Kingdom

Keywords:
risk management,
risk assessment,
risk analysis,
risk evaluation,
laboratory
information
management system,
LIMS,
laboratory
automation

This tutorial outlines some of the common risks that may be associated throughout the development and implementation of a laboratory automation project such as a laboratory information management system (LIMS) or another automation project. It presents a scheme for undertaking risk management to help assess and mitigate the degree of risk associated with each of these factors. In the case of high-risk factors, suggestions are presented to manage or help avoid the problem.

Risk management is an ongoing process. It begins at the start of a project and should be reassessed at intervals throughout the project to re-evaluate existing risks and to see if any factors have changed or new ones have emerged. (JALA 2004;9:72–86)

Introduction

There are many Laboratory Information Management System (LIMS) and laboratory automation projects that have collapsed or failed to deliver the expected benefits. Furthermore, surveys of information technology (IT) projects frequently show that many have run over budget, and nearly all projects end up with a changed specification from that originally described. When organizations used IT within laboratory areas in the 1980s and early 1990s, any failures were either covered up or written off as "one of those things" that should be put down to experience. Since then, organizations are far more cost conscious and sensitive of failed projects. Although failure is a powerful learning experience, it is usually never incorporated into a corporate knowledge base for use by similar projects in the future.[1]

A project is a single activity with a well-defined set of end results such as the successful implementation of a LIMS or another automation project. It follows a systems development life cycle (SDLC) from inception to completion.[2] A project does not exist in isolation and must often be coordinated or interfaced with other projects within the parent organization. Projects involve high levels of interdisciplinary communication and coordination with groups of specialists who are not usually used to such interaction. To aid the delivery of successful projects, project management provides an organization with the tools to plan, organize, implement, and control the activities necessary to achieve this.[3] This tutorial is intended to be complementary to existing project management techniques and methodologies.

The complexities and multidisciplinary nature of projects require that the many tasks and deliverable parts of each be put together so that the prime objectives of performance, timescales, and cost are achieved when delivering the defined project endpoint. There is a relationship between these three factors that has to be traded off by the project manager. Some of these tradeoffs can involve risk management in varying degrees. This tutorial aims to discuss some general risks and the management of them to ensure a successful outcome of an automation project and is a revision and update of an earlier paper on risk management by the author.[4]

Correspondence: R. D. McDowall, McDowall Consulting, 73 Murray Avenue, Bromley, Kent BR1 3DJ, UK; Phone & Fax: +44.(0)20.8313.0934; E-mail: R_D_McDowall@compuserve.com

1535-5535/$30.00

doi:10.1016/j.jala.2004.01.002

Background Reading

Although not directly referenced in this tutorial, the following books are useful background reading for computerized system failures (and the occasional success):

- *Crash: Learning from the World's Worst Computer Disasters.* Tony Collins with David Bicknell (1998), Simon and Schuster, London. ISBN 0-684-81687-3. The 10 deadly sins of computer failure are worth reading along with the case studies of many failed computer system projects—read and learn. However, as the authors note in the preface, the book has gone through two reprints and a second edition, but the book has not had the slightest beneficial effect.
- *Assessment and Control of Software Risks.* Capers Jones (1994), Yourden Press, Upper Saddle River, NJ. ISBN 0-13-741406-4. This books goes into more detail about project failure, but it is a more academic approach to the subject than *Crash.*
- *Patterns of Software Systems Failure and Success.* Capers Jones (1996), International Thompson Press, Boston, MA. ISBN 1-850-32804-8. Following the themes of his 1994 book, Jones looks at the reasons for successes and failures of software projects.
- *Managing Risk: Methods for Software Systems Development.* Elaine Hall (1998), Addison-Wesley, Reading, MA. ISBN 0-201-25592-8. A detailed approach for managing software development that can also be applied easily to automation and robotic projects.

Although the emphasis of these books is mainly on software, the principles outlined in them can be applied to other automation projects with little effort.

Project management is also important and plays a major role in determining if a project will be successful or not. Two key books that should be read are:

- *Software Project Management, A Unified Framework.* Walker Royce (1998), Addison-Wesley, Reading MA. ISBN 0-201-30958-0. Excellent book on managing software projects. Read and follow the principles and advice in this book and you won't need the next one.
- *Troubled IT Projects: Prevention and Turnaround.* John Smith (2001), Institute of Electrical Engineers, London. ISBN 0-85296-104-9. A good practical approach to project salvage and resurrection.

Risk Management

To overcome possible poor implementation or failure of a LIMS or laboratory automation project, risk management should be carried out at most stages of the system development life cycle. Risk management should be used in conjunction with project management techniques to manage the overall project. Therefore, identification of the risk factors should allow better management of a project and identify specific areas where additional expertise or care should be taken.

Definitions

Risk is defined for the purposes of this article as the chance or probability of an event occurring that may alter the progress or outcome of a LIMS or laboratory automation project.[4]

The following definitions are taken from ISO 14971:[5]

- Risk management is the systematic application of management policies, procedures, and practices to the tasks of analysing, evaluating, and controlling risk. From Figure 1, this is the overall process that is the subject of this tutorial.
- Risk assessment is the overall process of a risk analysis and risk evaluation. This is the major subprocess and comprises analysis and evaluation of risk as shown in Figure 1.

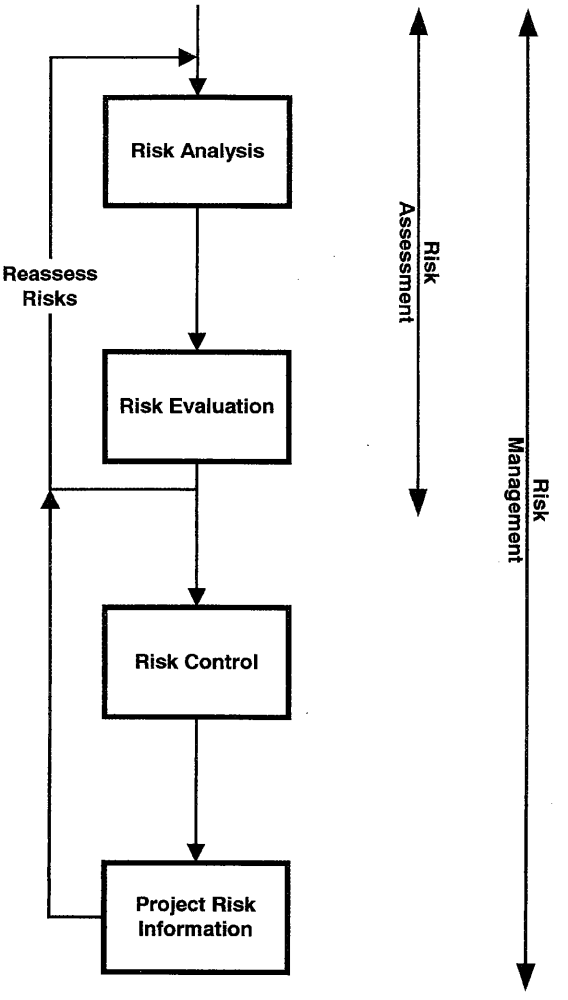

Figure 1. Risk Management Process Flow (Adapted from Ref. 5).

- Risk analysis is the systematic use of available information to identify hazards and estimate the risk.
- Risk evaluation is judgment, based on risk analysis, of whether a risk that is acceptable has been achieved in a given context.
- Risk control is the process through which decisions are reached and protective measures are implemented for reducing risks to, or maintaining risks within, acceptable levels.

Risk Management as a Process

There is little written in the scientific literature on risk management. Most risk analysis and management is intuitive and undertaken informally by project managers or project teams as a result of their experience or common sense. However, inexperienced individuals or project teams may have problems that could be mitigated or eliminated by the advance knowledge or experience of the common risks associated with LIMS and automation projects. The overall approach is shown in Figure 1.

Risk assessment and management is not a one-step operation but should be carried out at key stages of the SDLC of any project, and it is iterative. A project starts with a high degree of uncertainty and hence high risk. As it progresses, uncertainty in some areas is reduced but in others it can increase, hence the need for repeating the risk analysis and plan approaches to counter any newly identified risks.

Risk Analysis. At the top of Figure 1, the input should be at key stages of the project such as when a project definition document is written, system selection, or before implementation and rollout. From the project plan, the individual tasks can be identified for this portion of the work and analysed for potential risk factors. Using the knowledge and experience of the project team members, risk analysis can be carried out and potential risks identified; alternatively, some of the risk management tables in this tutorial can be used (see Tables 1–6).

For any stage of the project, the risk analysis process is to:

- Identify the known or foreseeable factors or hazards that could pose risk to the project, i.e., risk factors that can impact a project can be organizational, financial, or technological.
- Estimate the risks associated with each factor. For example, what could happen if the factor occurred and what would be the impact on the project?
- Estimate the probability of the risk occurring.

Risk Evaluation. The evaluation process is very simple—it asks the question, Does the risk need to be mitigated or not?

If the answer is "no," then the risk is accepted and nothing further is required. However, if there is a requirement for mitigation, then the risk moves into the next stage of the process: risk control. Typically, only high-risk factors will pass to the next stage; however, this decision depends on the criticality of the project in question.

Risk Control. Once the high-risk tasks have been highlighted, then it is possible to prepare plans and countermeasures to overcome the risk. "Risk Factors during Project Definition and Start-up," "Evaluation and Selection of the System," and "Risks Associated during Development and Rollout" discuss the risk and some of the possible approaches to mitigation. Note that it is not always possible to eliminate a risk, as this may be impossible or require too much effort; however, sufficient work needs to be done to ensure that the impact of the risk is managed and is acceptable.

The project manager then implements these approaches within the updated project plan. Milestones of the project

Table 1. Risk factors associated during project definition stages

Risk factor	Low risk	High risk
Project scope	Well defined	Undefined
Project deliverables	Well defined	Undefined
Benefits of the system	Quantified business impact	Undefined
System requirements	Straightforward, using standard components and technologies	Complex, using custom components and new technologies
Personnel providing application knowledge	Knowledge of both IT and user areas	Lack user area knowledge
Project members who have experience of business area	All project members	None
Status of documented procedures in user area	Complete and current	Nonexistent or outdated documentation
Number of computer applications that will interface with the system	None	Two or more computer applications
Status of these applications	Operational	Under development
Resource required	<5 resource years	>15 resource years
Time to develop the system	<12 months	>3 years

Table 2. Risk factors associated with system sponsorship and user commitment

Risk factor	Low risk	High risk
Project sponsor	Identified and has a strong user influence	Unknown
Attitude of user management	Fully support the project	Resistant and skeptical
Attitude of the users	Understand and support the project	Resistant and skeptical
Organizational maturity	Able to use the system effectively	Unable to understand rationale for the system or use it
Relationship of the project to the strategic IT plan	Included in the plan	Not included in the plan

can be identified and progress of the project reviewed against these; the same is true of risk factors. Once the progress of the project has been evaluated, this can be fed back into the system for an updated risk analysis. As can be seen, risk analysis is linked very closely with project management, and the two approaches should operate intimately.

Project Risk Information. As the project progresses, a body of information is collected. It describes how the project risk has been managed and how effective the approaches have been. It is important that this information is not forgotten or ignored. Reuse, cross-reference, and update the information; it can be used to feed back into the risk cycles as shown in Figure 1. Also, experience from failed projects, discussed briefly in "Learning from Failure," should also be incorporated in the organization's experience of automation projects as lessons to be learnt for the future.

Areas of Risk in the System Development Life Cycle

As laboratories depend so much on automation, and LIMS in particular, it is essential that management, users, and the project team should do as much as possible to minimize the risks to ensure a successful implementation. If an LIMS is not functioning effectively, then work within the laboratories will be disrupted and productivity will suffer. However, when an LIMS is operating badly or not working at all, then the customer does not obtain the information and the reputation of the laboratory suffers.

The three main areas of risk within the SDLC are:

- Project definition and start-up (see "Risk Factors during Project Definition and Start-up")
- Evaluation and selection of the proposed system (see "Evaluation and Selection of the System")
- Implementation of the system (see "Risks Associated During Development and Rollout")

Table 3. Risk factors associated with the impact of the system on the organization

Risk factor	Low risk	High risk
The new system	N/A	Replaces an existing one A completely new system
Effect of the system on the IT department of the organization	Little change	Much change
Procedural changes required by new system	Little change	Much change
Organizational structures	None required or no changes to existing ones	Not considered
Policy changes required to support the new system	None	Extensive

Risk Factor Tables

The factors highlighted in the accompanying tables will allow continuing assessments of risk to be made of individual projects at various stages of the SDLC. As there is usually an underestimate made of the technical complexity of systems development, risk can never be eliminated totally from a project. However, the more that significant factors can be contained or contingency plans made to manage them at the start and throughout a project, then, the greater the chances of success. Over optimism, especially in the planning stage, is a chronic problem, resulting in projects being over time and over budget. Therefore, contingency time and money should be included in all plans.

Risk can be assessed as probable (high risk), possible (low risk), and improbable (negligible). Another approach is to assign to each factor a numerical value, say 10, for the highest risk, and 1, for negligible risk. Risk can now be evaluated on a continuum, which can be useful for an assessment of risk for a number of projects such as a prioritization exercise. However, in the tables presented in this tutorial, only high and low risks have been evaluated, as it is preferable, in the author's view, to keep the scheme as simple as possible.

Risk Factors during Project Definition and Startup

The risk factors that may be encountered during the start-up phase of a project can be divided into four main areas:

1. Project definition
2. Sponsorship of the system and user commitment

3. Impact of the system on the organization
4. Management of the project

Each is discussed in more detail below. This section is the longest in this tutorial since, if this is not defined and agreed at the start, the whole project is worthless and has a low probability of success.

Risk Management during Project Definition

Outlined in Table 1 are some of the key issues that should be considered for risk assessment and management during the proposal definition and writing of any automation project. The areas of low and high risk (relating to possible and probable risk, respectively) are highlighted in the two right-hand columns for each factor. Every factor is considered below with suggested ways of managing the risk posed. The main effort in this phase of a project is commonsense management. There are no excuses.

Project Definition and Deliverables

Before starting a project, it is common sense to define the overall scope and tasks the new system will replace or carry out. It is important for all concerned that this is achieved in a project proposal or definition document. The content should explain, in non-technical language, what the system is to achieve when it is delivered. As this is the baseline for all future work on the project, it is an essential deliverable.

Moreover, the users and management must accept and underwrite the content of this document. The alternative is a poorly defined project with no focus. Thus, it is easy to introduce trivial or non-essential functions at the whim on an individual, which can waste time and effort, or worse, functions with little practical use. Moreover, a poorly defined project can select the wrong equipment or application to meet business needs.

Table 4. Risk factors associated with project management

Risk factor	Low risk	High risk
Experience of project manager	3 or more projects	No prior experience
Managing the project	Full time	Part time
Project team assigned	Full time	Part time
Experience of team members as a team	All worked together before	All are strangers
Number of times application/ system implemented	More than once	No experience of this application
Team location	In one place	In several locations

The deliverables expected throughout the SDLC should be outlined at the start of the project to avoid not meeting user, management, and, where appropriate, regulatory expectations. Therefore, time should be spent discussing with the users, management, and especially the project team members the importance of any deliverables. Within a regulatory environment, these deliverables will form the basis of the quality development and validation of any automated system.

Defined Business Benefits. To avoid premature cancellation of the project due to budget cuts or management change, define and, if possible, quantify the business benefits that the system is anticipated to deliver. Senior management will need this information for project approval if over a preset spending limit, and continued management support is essential for longer-term projects. The advantages or direct benefit to the organization and the user should be outlined. To obtain the information to write this authoritatively,

Table 5. Risk factors associated with system selection

Risk factor	Low risk	High risk
New or non standard hardware or system software required	No	Yes
Team has experience of this application/system	Expertise available	Used for the first time
New language(s) required by the project	None	Used for the first time
Database used in the application	Well established in the organization	New in the organization
The system processing	Batch processing	Distributed system
On-line response required	>7 seconds 90% of time	2 seconds or less 90% of time
System availability	95%	>99%
Technology mix (database, network, etc)	Existing or simple architecture	New or complex architecture
Team's knowledge of the package	Previous experience	No knowledge
Organization has worked with the vendor	Three or more times before	Never
The package matches the system requirements	Well (little customization required)	Poor (major customization required)
Computer department involvement in package selection	High involvement	Not involved
User department involvement in package selection	High involvement	Not involved
Vendor reputation	Good	Poor or unknown

Table 6. Risk factors associated with system development and implementation

Risk factor	Low risk	High risk
System scope fixed and prioritized	Yes	No
Changes in organization or policy implemented	Yes	No
Change control procedures in place	Yes	No
Scope matches existing or proposed working practices	Yes	No
Documented roll-out plan available, detailing who, when and where	Yes	No
Sympathetic users identified for prototyping and testing	Yes	No
Quality Assurance involved for pre-operation audit and GMP advice	Yes	No
Development documentation to be produced identified	Yes	No
Performance of development system matches expectation	Yes	No
Contingency plan available if performance not good enough	Yes	No
Focused training planned and agreed with the users	Yes	No
Documentation available for users, backup and support staff	Yes	No
Assistance arranged for new users in period after training	Yes	No
Work schedules altered to cope with post training productivity loss	Yes	No
Plan for management of user expectations	Yes	No

involve experienced staff from the user environment as well as laboratory customers.

The business benefits are useful for defining, in another way, the target of the system to be developed. This definition can be of positive use in helping to make decisions concerning which functions are to be evaluated during selection and development.

System Requirements and Documentation of Current Procedures. At the start of a project, the system requirements are relatively vague and can hide a number of complex technical, procedural, and even organizational issues within them. However, even at this early stage, the requirements of a project and the operations carried out within the target laboratory should give an understanding of the degree of complexity involved. If the system to be implemented appears to be complex, a number of approaches to reduce the risk can be suggested:

- A complex system could benefit from a detailed systems analysis to understand the information and data inputs, internal operations, and outputs. This should give a better understanding of the requirements and may help the new system support decision-making.
- As users often find it difficult to explain exactly what functions they do or are uncertain what they want the system to do, this may suggest a prototyping development with engineering or software projects. This approach should help the users debate and develop what they require and reduce the risk of the overall project, as the target can be scoped and the functions defined based on the experience of the prototype.
- A review of the working practices of the laboratory should also reveal if the processes undertaken should be changed prior to the introduction of a new system. One area where this is important if electronic signatures are to be used: an electronic process has to be defined, as the existing paper-based process will be inefficient.[6]

If current procedures are documented, these will help define the current practices and system. In contrast, poor, outdated, or no documentation can cause assumptions, perhaps wrong ones, to be drawn and requirements defined from incorrect information. Great care must also be taken not to assume that even if practices are defined, say in standard operating procedures (SOPs), that the current working practices in the laboratory match them. No assumptions should be made in this respect, as these documents may be major works of artistic fiction. Enlist the help of expert users to help define the current system.

It is essential to define current working practices, modify them where necessary, and map them onto a proposed system to help selection.

Knowledge of the Project Team Members and Users. The skills of all of the project team members should be assessed before the start of the project. Clearly, where the IT and laboratory automation members have been involved successfully with similar projects in the past, especially within the same area, there will a high degree of confidence and technical skill. In contrast, a team that is new and has little experience will require team building and technical training, ideally, before the project starts.

Equally, the experience of the user representatives working with the project team members may not be adequate to get a high-performance team working immediately. Therefore, some on-the-job training in project team membership may be appropriate.

An issue of major concern is the degree and depth of understanding and knowledge each member has in the other team members' disciplines. When there is none, a level of common understanding has to be developed. Equally important is the degree of knowledge and understanding the computer staff has in the laboratory. In the author's experience, this understanding takes much effort to acquire but is a worthwhile investment. A corollary is that carefully trained IT staff must be retained by the project; otherwise, momentum will be lost. To reduce risk before the project is

fully underway, management has the responsibility to ensure that the team is formed and has the required level of knowledge and understanding to do the job.

Interfaces to Other Systems. This aspect is not always identified in project proposals, but for integration to form an efficient organization, IT and automation projects should not exist in a vacuum but interface with each other to provide an integrated information environment. If the system is a standalone, no interfaces with other systems are deemed necessary, and development can proceed unhindered. In contrast, if the system has to interface with one or more systems, this adds complexity and risk. The interfaces, especially the data inputs and outputs, must be carefully defined and documented along with the responsibilities of who does what. This is acceptable if the systems already exist and are functional, as the interfaces are tangible.

Problems can arise if the interfaces are with proposed or partially developed systems, since interfacing with these applications increases the risk assessment. Now, additional time is required to identify where the projects overlap and how they should interface; liaison between projects is essential. Liaison may include sharing project team members, planning inter-project dependencies, or identifying the other project's deliverables.

Resources for Project Development. Smaller projects carry less risk. Therefore, to minimize the risk for larger projects, there are a number of measures that can be used to reduce the risk to acceptable proportions:

- Ensure formal project planning and monitoring with clearly identified deliverables and milestones, although this can be time consuming, and project team members could lose enthusiasm for the project. It is essential to focus members on the original aim of the project.
- Large projects can be broken down into smaller ones with discrete endpoints. These smaller projects, when complete and aggregated together, constitute the overall system. Alternatives are to reduce the original project scope and produce a minimum working system that can prove its effectiveness before additional functions are added or by using a phased development approach.
- Large projects developed over long time periods can cause problems in maintaining enthusiasm and user commitment. Moreover, any organizational changes could result in changes in sponsorship and less commitment or resources for the project.
- Large projects could justify the use of application development tools, such as computer-aided software engineering (CASE), to enhance productivity and decrease the time taken for various phases of the project. If the team has used this approach successfully in the past, this would be beneficial to the project. However, if this approach means introducing new technology, it may increase risk instead of reducing it.

Project Timescales. In today's organizations, a timescale of 2 to 3 years is a long time; in some companies, this can be the expectancy of an organizational structure or time between mergers. Therefore, if the timescale exceeds this, the project is unlikely to complete before the next change and is very unlikely to bring business benefit. Therefore, projects that last longer than 18 to 24 months are high risk. As described above, reduce the scope or break the project into a number of clearly defined phases. However, in doing this, it is essential that each phase provides defined and quantifiable business benefits of itself, or it is not work doing.

Risk Factors Associated with Sponsorship and Commitment

Presented in Table 2, and described below, are the risk factors associated with sponsorship of the proposed system and the commitment of the users to accept it. Although presented here as factors associated with the start-up of a project, they must be reassessed during the project in the light of any changes in senior personnel, organizational rearrangements, and influences on the users. This is especially so in a project that has a long timescale or has been delayed.

Sponsorship of the Project. The best way of identifying a project sponsor is to ask the question, Who provides the money? Active sponsorship of large projects is important to persuade people to use the new system. A sponsor who is just a figurehead is a green light to wasting a large amount of money, as there will be few questions asked if the project fails to deliver. Senior management's understanding of automation and computer projects is usually based on two premises:

- The project is expensive
- It won't work

These perceptions must change.

If the project sponsor is not strong, political battles within the organization units under this individual can result in project delays due to a lack of decision or management commitment, especially in large projects. Therefore, to avoid these problems, procedures for resolving disputes should be instigated by the user management.

Attitude and Commitment of User Management. The attitude and commitment of the user management is essential for the success of any project. Apparent lack of commitment may indicate that they are unaware of the potential benefits that the system may bring or of plans to change the laboratory's direction. The manager of the user area should be briefed regarding the benefits that the system should deliver and its ability to enhance the business objectives of the specified area.

Winning the hearts and minds of user management is one option. If the project sponsor links a performance bonus to a successful project implementation, this adds the dimension of the wallet or purse to the equation.

Attitude of the Users. Even with total commitment of the project sponsor and user management, users can cause serious problems throughout the whole project by refusing to cooperate during all stages of development. This may be the result of fears of radical change that would result from the operation of the system. Mechanisms for effective communication to representatives of the user community need to be established by regular status reports or meetings that should be continued throughout the development cycle. Concerns of the users should be communicated to the project team and management. If there are organizational impacts of the system (see "Changes in Organizational Structure"), these should be identified and communicated to the users and discussed to obtain a consensus agreement.

The maturity of a user organization to support a LIMS effectively is a factor to be considered early in a project. If, in the judgement of the project team or user management, the organization is not capable of supporting an automated system, then an education programme should be undertaken for the whole user base.

Organizational Maturity. Not often considered when assessing the risks of a laboratory automation or IT project is the capability of the organization to implement the new system. This will vary, but a simple way to find out is to review how successful previous projects of this type have been. Projects that have been an outstanding success, rare but they do occur, will be a pointer towards the successful design and exploitation of automation or IT. It may also indicate that risk taking is encouraged, although this needs to be established by direct evidence of a corporate policy or equivalent.

More often, if this is the third or fourth attempt at a project, it will indicate a poor track record and, therefore, a high-risk project that has to be handled more carefully and in a risk-averse manner.

Relation of the Project to the Strategic Plan. If a project falls outside of the scope of a strategic automation or IT plan, then the risk increases as the obvious question arises, Why is this project important? Conflicts may result from an unplanned project being given priority and resources over existing ones. It is best to find out the reasons for a new system that is outside of an existing plan. If there has been a change in the business strategy, then the IT strategy should be revised accordingly. The place of the new project and any dependencies among other projects should be identified. A strong and committed project sponsor may be required.

Risk Associated with the Impact of the System

Described below and presented in Table 3 are some of the risk factors to be considered when examining the impact of the new system on an organization.

New or Replacement System? Regardless of whether a new or replacement is considered, both present a high risk; however, the nature of the risks are slightly different and are discussed here Both approaches impact the user community by requiring it to change; it is the issue of change that presents the risk and, thus, it must be effectively managed.

A replacement system should not present too many problems with respect to the impact on the organization, if, and only if, the replacement simply automates what was automated in the previous one. However, many IT applications do not fully automate the process, and automating the status quo simply perpetuates the problem. Replacement of an existing system should be undertaken as enumerated here:

- Understand the strengths and advantages of the current system, as these have to be maintained in the new system.
- Understand the weaknesses and disadvantages of the current system and ensure as many of these are overcome in the new system—this is the enticement for the current user community to migrate to the new system.
- Evaluate the current scope and boundaries of the system. Do they reflect current business needs?
- Review how the system is used (e.g., fully electronic versus manual input from paper printouts) and see what improvements can be made.

Design the replacement system with the current business needs and processes in mind, not the old ones. However, a new system tends to impact many areas, including:

- Changes to existing ways of working
- Time to learn to use the new system effectively
- Computer or engineering skills to use the new system
- Organizational maturity to use the system
- Staff might be unsure of their duties and responsibilities when the system becomes operational or they could resist its introduction

To counter these impacts, management should assess the effects of the new system on the organization and the users. Communication of the benefits of the new system to the users should be undertaken, but remember to keep the statements realistic to manage user expectations. This discussion can be achieved in groups or individually.

Involvement of the users in all stages of the project is essential. Areas where this can occur are project planning, analysis, testing prototypes, and implementation. Request input on how to structure and phase the training to use the new system. A champion or champions for the system should be identified and involved throughout the project.

Impact of the System on the IT Department. As systems become highly integrated environments and work in close cooperation over networks, a new system can have different levels of impact. If there is little change in the operation and management of the computer, there will be few problems apart from negotiating the facilities management contract. At the other end of the spectrum, there must be enough capacity or bandwidth on the network to accommodate the anticipated data flows from the laboratory to the server plus

sufficient workstations and peripheral devices to access the system. In short, ensure there is sufficient capacity for the new system to operate effectively.

If the IT department has no experience with newly acquired hardware, operating systems, and/or application software, this will have an impact on the support staff and will increase risk accordingly. Organization members should have the skills and experience to run these new methodologies efficiently. If not, they will have to be acquired by training existing staff or recruiting new personnel. The input from operations staff to the project team during evaluation and selection can identify many of these areas. To reduce the risk and aid communication between various applications, the first intent should be to purchase or develop a system that is consistent with the current systems in place within the organization. This aspect will be considered in more detail in "New or Non-Standard System Components." The author would advocate using corporate standards whenever possible, as this will be the easiest to implement.

Documentation of system procedures, coupled with effective training, to use any new hardware and operating systems must be in place before the system goes live.

Procedural Changes Required by the New System. Project risk can be greatly increased with the failure to recognize early in a project the need for any new or revised procedures; thus, plan and implement them rapidly. The current working practices should be reviewed, and the level of the user's commitment to change procedures should be established. If change is resisted, do not implement change via the computer system but change procedures, if possible, by altering the manual ones first. This is a relatively cost-effective route to take, as small modifications can be undertaken easily and rapidly until the new manual procedure is streamlined and effective. Then, overlay the new system on top of the modified manual system. In this way, problems can be resolved with the new procedure without the computer being used as a scapegoat by dissatisfied users.

Changes in Organizational Structure. Computer systems have the power to cross functional and organizational boundaries with ease. The failure to recognize and plan for any changes may result in staff not knowing new responsibilities or roles or in disruption occurring from reorganization of organizational units; hence, there will be an increase in risk to the project.

The impact of any organizational changes should be documented clearly during design and development, although it may be alluded to in the project proposal wherever possible. There should be change management of any such changes over a specified time period. Always, if possible, allow time for the changes to settle down before implementing the new application to avoid too much change in a short time period. Again, the communication of the realistic benefits of the new system to the users should be undertaken.

Policy Changes to Accommodate the New System. Changes to policies should be identified and controlled by the user management. Since these are not always identified until the detailed design stage or the development stage, any delay in implementing these could delay the operation of the project. The resolution of these policy issues must be made before development can take place.

Policy changes may be the result of the introduction of new technology, organizational changes, or modification of procedures caused by the new system. Therefore, it is important to identify and resolve any policy changes rapidly but not before considering the impact of the changes. If large numbers of policy changes have to be made, there should be a mechanism in place to document and inform all staff of them—a user appointed as a coordinator might be one approach to use here.

Risk Factors Associated with Management of the Project

Presented in Table 4 are the common problems that can give rise to risk when assessing the approach to project management and the membership of the project team.

Experience of the Project Manager. An inexperienced project manager may have difficulty developing an efficient project plan and modifying that plan as the project progresses. Moreover, not all tasks may be identified, or the project plan may not be broken down to a sufficient level to enable accurate scheduling. Taken together, this often results in delays to the project and missed deadlines, with tasks rescheduled or additional ones included, often at short notice. The impact can be damaging to the project, as budgets may be increased, and there may be loss of confidence by the users or cancellation of the project, as benefits have not been obtained in a timely manner.

To counter this, the project manager should be trained in project management techniques. When drawing up the plan, allocate more time for the completion of the tasks to allow for slippage, or allow slack time. Regular reviews of project progress should be set up. To gain from the experience of others, read the status reports and reviews of similar projects completed within the organization.

Full-Time Project Manager. Depending on the size of the project and the resource available, it is preferable to have a full-time project manager. This avoids conflicts of interest with line management responsibilities and allows the ability to focus on key issues that might not occur if it were a part-time position. In some respects, this is a management decision about the amount of time and resources allocated to a project. However, there is also the onus on the project manager to inform management if he feels overworked when given dual responsibilities.

Full-Time Project Team. Whilst it is common for the project manager to be allocated full time to the project, the team

members are usually allocated on a part-time basis. Here, the line/matrix conflicts outlined in the last section will become apparent as the project competes with the line for the resources of these skilled individuals. In this situation, errors can be made or delays occur that could impact on the project, ordinary work, or both.

To manage this situation effectively, it is important to define accurately the amount of time that a project team member should spend on his respective duties. This will reduce the amount of time available for line work, and, accordingly, the manager of the individuals should negotiate with clients regarding deadlines involving these staff. If specific tasks for the project such as in-house evaluation require an individual's time, this must be negotiated with the supervisor well in advance of the event.

Ideally, the project team member's position description should be changed to reflect the work done on the project so that both the line manager and the project manager can evaluate the individual's overall performance.

Project Members Operating as a Team. When the project team is composed of members who have not worked together before, some delays may occur in the initial stages of a project. Team members need time to get to know other member's personalities, understand their skills, strengths, and weaknesses, and learn how to work together. Risk arises if the team lacks skills or understanding of the technology involved or the knowledge to complete the project successfully.

To overcome this and reduce the risk, attempt to use staff that has worked together as a team. Working to the strengths of an individual is always preferable to training another member. However, this approach carries the risk that IT and automation skills can often be in short supply, and one individual can often be carrying out several tasks, which can conflict with line duties. A way to transfer skills should be included when feasible and when time and resources allow.

Experience with the Application or System. Often, the majority of project team members from the user areas have little or no experience with a new type of application. Without experience, the team will not have the insight to avoid mistakes or enter blind alleys. Additional time may be required for reviews and revisions of the plan and its execution.

If similar projects have been introduced in the organization, utilize the knowledge from some of the team members as an internal consultancy role. Ensure that more time is allocated to the project to allow for problems.

Multisite Projects. The concept and introduction of a corporate LIMS or an automation project may involve two or more sites. Whilst from the corporate viewpoint, this is an effective use of resources for development and maintenance, and the benefits will be significant over the development of different point solutions at every site, problems will be encountered. By its very nature, a project covering more than one site tends to be larger, more complex, and hence more expensive than one at a single site. Senior management should look carefully at these projects, as a significant proportion of their IT budgets will be involved.

Communication may be difficult, especially if there are time differences involved that are greater than one or two hours. This can be overcome by the use of electronic mail facilities or an Intranet site for common-interest items. Progress updates will need to be regular for all sites and held centrally at one location to ensure control of the overall project.

There may be lack of coordination at the sites where the project manager is not located. Travel, often extensive, will be involved for the manager and several key members. Budgeting of money for this and the associated subsistence is essential. Different sites may have different working practices, policies, and organizations. These typical issues, raised by a standard system, will have to be resolved at the senior management level before much progress can be made. In companies working on a global basis, there will also be cultural differences, methods of working, statutory holidays, and even the length of the lunch break to be taken into account. The timescales for the project will need to be increased to account for these factors. However, the overall benefits to the organization should outweigh these difficulties.

Evaluation and Selection of the System

In this section, it has been assumed that a commercial LIMS or laboratory automation software package is being selected. However, if an in-house system is being developed from components that the organization will integrate and develop, a few modifications to Table 5 are necessary. The general factors involved, such as the technology used in the system, the mode of selection, and the impact of the vendor, are discussed separately.

Technology Components

Presented below are factors involving the technical components of a system that influence the risk during the selection process.

New or Non-Standard System Components. Increased risk to a project will be incurred if new or non-standard system components are selected for the application. Under this category are included:

- Hardware and operating system
- Networking protocols or components
- Application software, including languages, databases, tools, techniques, or utilities

The risk in selection of non-standard components is manifested in several ways. The development team and the support staff need to become familiar with the respective

components. This may require training that may be extensive as well as costly. The extent of integration between these new components and any existing applications may raise technical problems at the very least. Training to use these packages, and potential delays due to technical problems to be solved, must be an essential element of the project plan.

Establishing contact with the vendor's technical experts for specialist information and advice may be a way of gaining information to reduce risk or to obtain solutions to actual problems experienced. It is preferable to keep to the corporate standards wherever they are established for easier implementation and maintenance.

The choice of packages that do not conform to corporate guidelines must be made carefully:

- Does the database have sufficient flexibility to undertake the tasks now and in the future?
- Is the application development language suitable for the task?

The choice of the wrong database or development language will have a major impact on the project's ability to deliver the expected benefits.

Type of System and Processing. The greater the complexity of the system, the higher the risk that something will go wrong—this is Murphy's Law of laboratory automation. Management of risk approaches should be adopted to choose the simplest approach consistent with supporting the application effectively. The choice of a pilot system to size the processor, memory, and disc I/O accurately before the installation of an operational system may be one avenue to take. If distributed processing is required, implementing core functions in two locations first could be adopted and preferable to finding the completed package does not work as anticipated. Some applications may require on-line data capture in real time; this requirement may entail having a failure resistant hardware configuration. The need and justification for every requirement should be investigated thoroughly.

Response Time. The faster the response time required by the application and the users, the higher the risk. Failure to meet this performance criterion may result in loss of user involvement and interest. The sizing of hardware components and the design for rapid database searches for urgently required data may be crucial, but remember that not all data may be needed rapidly. Ensure that the computer has the expansion capacity for the next three to five years to cope with increased demand, either with sufficient capacity purchased at first intent or by planned incremental growth.

System Availability. The need for high system availability should be investigated and justified; there is often a stated requirement for 24/7 availability but few systems actually require it (best justified are manufacturing processes for raw materials and active ingredients).

If a high degree of availability is placed upon the system, the supporting hardware and network also need a high level of availability. Therefore, fault-tolerant hardware may be justified in cases of near-100% availability being required. Procedures for identifying and solving problems should be developed, as should effective and rapid contingency plans and disaster recovery procedures—e.g., consideration should be given to spare hardware systems being available to be started up in the case of an unplanned system or IT infrastructure failure.

Technology Mix. The greater the number of technologies that have to be integrated into an application environment, the greater the risk becomes. Wherever possible, keep to the simplest approach that is consistent with the requirements of the application and that meets the needs of the user and organization. Wherever possible, use components, or proven technology, that the organization has knowledge of and has used successfully before. Here, the success rate of the organization in implementing IT and automation projects plays a role. An organization with a successful and innovative track record in implementing projects can probably justify the risk involved with a range of technologies. However, a less-evolved organization should lower its sights and err on the side of caution.

Risks Associated with the System Selection

Selecting the System. When a system or an application package is chosen, there are a number of parties with vested interests within an organization such as the IT Department, the user laboratory, and, in a regulated industry, the quality assurance unit (QAU). Using an effective project team approach, all parties should be represented and have input.

From the IT departments viewpoint, the users may have chosen the wrong package for a number of reasons such as non-standard components, new technology, or the database does not appear to fit the requirements. The input of the IT department should be to check the requirements and package to assess the degree of fit from the IT perspective. It is possible that the package may not meet user expectations, or it may take considerably longer to implement than anticipated from an IT perspective. This can be resolved by ensuring that the package is tested fully with tests that represent functions carried out by the users.

If the IT department, with little input from the users, selects the package, the greatest risk is that the users will reject the system. The users know their own environment the best and appreciate the functions they require. The QAU interest is that the selected system can be validated, and they can use the system to carry out audits effectively.

The closer the package matches the requirements, the less risk incurred by the project. The further the package is from the requirements, the more customization will be required, or the users will have to modify their working practices to use the package. Both instances increase the risk of the project

and can lead to excessive time delays or user rejection. It may be appropriate in this instance to consider a custom-designed system rather than a package. Alternatively, redefine the scope of the project and reassess the fit to the modified requirements.

Seduction by Technology. Evaluating a system or application package without a system or user requirements specification is asking for trouble. There is no baseline from which to make a value judgement and is likened by the author to being seduced by technology—it is unknown if the system can actually meet the business needs, as they have not been documented. Therefore, before evaluating systems, it is important to document the requirements and have objective means of evaluating the systems being reviewed.

The Vendor. The project will have increased risk if the organization has no experience dealing with a specific vendor. Without first-hand knowledge of their contract negotiating techniques and their willingness to modify their system (if required), a laboratory could end up with a system that does not support the business and that incurs expensive delays. This risk is increased if the company is new or has only a relatively small number of installations.

To a certain extent, an indication of a vendor's attitude toward existing customers and their problems can be obtained from site visits. However, it is important to remember that a vendor will not usually take potential customers to a site at which they have had many problems. The site most likely to be selected will be one with which the vendor has a positive relationship.

To counter this, it may be prudent to insist that all agreements with the vendor are in writing; this may also be true of statements made by sales personnel who are attempting to win an order. Access to the vendor's technical specialists can build confidence in dealing with a vendor and be the start of a good working relationship. Communications, both formal and informal, should be established, and any issues discussed should be entered into a log as a formal record of progress.

Vendor Failure. This risk covers the failure of a vendor due to either commercial failure of the company or, more often, the withdrawal from a market sector for commercial reasons or changes in business direction. If any of these problems occur, it is important that the organization does not suffer a loss of its investment—both money and time. Contingency plans may be drawn up for the maintenance of the system, at least until a replacement can be found and implemented (possibly a one- to three-year period).

To try and avoid failure of this type before any order has been placed, and preferably during the selection process itself, obtain financial statements from each vendor under consideration. Non-disclosure agreements may be essential to obtain information, especially if it is not part of a published annual report. Key indicators are the length of time the vendor has been in the business and the growth of the company over that period. Within the IT area, many companies may have relatively short track records and may be relatively small. The impact that their product or products have realized in the time they have been available could be used instead. When considering automation, it is more likely that the company will be larger, but this is no guarantee of minimized risk, as some of the largest companies have changed direction and left customers with little or no future direction.

While care is needed in vendor selection, a track record and growth with a successful product is ideal, but these factors should not be used as exclusion criteria against smaller companies that may be emerging with a superior product.

Safeguarding the investment can be achieved by the use of clauses in the purchase contract—for example, all software and documentation should be provided or put into escrow with a third party in the event of failure. Access to source code is a contentious issue, but in the event of corporate failure, this may be the only way of maintaining the system. To protect the laboratory, it may be prudent to include a clause allowing the maintenance of the software by a third party if the vendor cannot or will not fulfil the contract. Incorporating items such as those outlined above is a long and complex process that should be undertaken carefully.

Make or Buy a System?

The best approach to minimize risk is to buy a commercial system rather than make or program your own. This preferred approach means that system development and maintenance costs are spread over the whole customer base, and there is usually a development of the system with new features being added, especially in competitive market segments of laboratory automation. However, often a system does not match exactly a laboratory's requirements, and here is the beginning of compromise: Do I change my ways of working (cheaper and better in the long run) or change the way the system works (short-term gain but costs more in the long run, as the laboratory upgrades from different versions)?

Many laboratory automation projects are custom or bespoke (unique and built specifically for a single group). These benefits are a tailored approach that matches the current ways of working but are expensive (the laboratory meets the full development, maintenance, and support costs). This is high risk. Often, the ego of the organization drives these projects, as there is sufficient money and resources to fund the work, but it will usually take longer than a commercial system. Custom projects are only truly justified when there are no commercial offerings.

Risks Associated During Development and Rollout

Of all areas of the SDLC, development and implementation are the stages that have the highest risk associated with them.

To a certain extent, a project can cope with poor sponsorship or the suboptimal selection of a system. However, the development and implementation phases are where the majority of projects fail. Even a technically perfect system that matches user needs can be lost by user indifference or hostility. Some common risk factors that could occur during development and implementation are presented in Table 6.

There are some factors that are unique to development and implementation. However, it is also the part of the project where many of the earlier risks will have their full effect if they have not been managed properly.

Fixed-System Scope

By the time the development of the system starts, it is imperative that the scope of the system is fixed and the functions to be customized are prioritized and agreed upon by the user management. If the scope is not fixed, users or managers could add additional functions without control: this is one of the major reasons for the failure of many projects. This could have a number of results; almost certainly, the system will be delayed and the functions added may not produce any meaningful business benefit. During any implementation, the core laboratory functions should be configured first. Additional functions must only be added later according to business need and under change control.

System Scope Matches Laboratory Working Practices. When the development starts, the scope should either match the working practices in the laboratory, or changes in the manual practices have been instigated so that they match the new system functions. System credibility can be lost easily amongst the users by an unplanned mismatch of system and working practices. Liaison amongst the user representatives on the project team with the system developers should help to alleviate this problem.

Change-Control Procedures. Once the scope has been fixed, there should be change-control procedures set up to debate and approve any additions, deletions, or modifications to the scope. Without change control procedures in place, there is a significant risk in uncontrolled development of a system.

The change control process involves a set of procedures and a review group. The latter can be either a subgroup of the project team or a separate group whose purpose is to review and prioritize any modification of the scope. Submissions, detailing the changes to be made, should be in writing with the business benefit laid out. Change control should avoid the trivial functions being added at the expense of more urgent ones, thus delaying the project. The corollary is that occasionally some important functions are missed from a specification, and this mechanism provides the means to have them authorized for inclusion.

Implementation of change control is also useful when the system is fully operational, as all changes to the system configuration should be proposed and authorized in this manner.

Documentation. The documentation of the system is a key quality issue. The main document required for the development is the scope describing the functions to be customized. Additionally, an outline of the change-control procedures, draft testing and validation plans, draft procedures for start-up and shut down of the system, and outlines for the user manuals are needed.

Documentation is required for validation of a system, but more importantly, it is essential for the smooth transition from development to operation. The time required writing good quality user and support documentation is usually longer than anticipated. Therefore, the tasks should be started well in advance of when they are required, and enough time should be allowed for the job to be completed, with sufficient quality to be the first line of support for the system.

Involvement of Users in Prototyping and Testing. Before development starts, the project team should have identified a group of sympathetic users who will be used to test prototypes or functions developed via conventional programming. The users should represent all groups within the laboratory environment. Note the use of the word "sympathetic." Credibility is easily lost during development by word of mouth and by the actual performance of a system. There is little point in selecting a group of users who do not want the system to succeed or who are skeptical toward the use of automation. What is required is constructive comment and criticism that will allow development of functions to proceed without detrimental comments about the system being made.

Implementation and Rollout

Detailed planning and availability of personnel in this phase of the SDLC are crucial to the credibility of management and success of the overall implementation.

Detailed Implementation Plan Available. Before commencing this phase of work, a detailed plan covering the implementation must be available. Details covered should be as follows:

- The implementation style for an LIMS should be clearly defined, and the implications of each approach thought through before starting.
- Training, which can be carried out in various ways (e.g., internal or external), should be planned and costed. Any external staff from a vendor should be informed of when and where they are required.
- External groups who submit work to the laboratory should know when training takes place and the impact of this on the work schedules. The latter should be rearranged to include the immediate post implementation period when productivity will be lower than normal.

The aim of the plan is to remove most of the uncertainty involved during implementation and direct resources to where they are most needed and when they are most required.

Training Plans Agreed. Once the implementation style has been agreed upon, the training schedule can be developed relatively easily. In the implementation plan, the groups of workers who will be trained to use the system and the order of training should be identified along with the support staff who must be on hand to augment training and solve any problems. Obviously, risk increases dramatically if staff are not trained to use the system.

Many vendors offer standard courses; however, this may not meet the needs of users where the system has been customised from the core system offered for sale. It may be beneficial to consider customizing training courses and holding them on site if there is sufficient demand to do so or the cost benefit is good.

Training is an easy target when it comes to budget cuts; instead of training all system users, only key ones are trained, with the aim of cascading the training from one or two key users to the rest of the user community. This is often a false economy, as the key users may be technically very capable but not professional trainers; skills and knowledge may not be transferred effectively to the key users who are poorly equipped to transfer what they have retained to others. Ensure training is properly budgeted and professionally carried out to guarantee that the organization has a good opportunity to gain the best benefit from its investment in the system.

Implementation Delays. There are a number of possible causes of delays including vendor failure to deliver a package within an agreed time frame or the writing of in-house software is slower than expected. More problematical are instances where the functions of the system do not match the current working practices in the laboratory, necessitating a delay to rewrite software. The lack of suitably qualified staff, either in-house or from a vendor, may impact the project at a crucial time. Regardless of the cause, delays in implementation are frustrating and have a bad effect on morale and a negative impact on the credibility of the system.

Using staff to work on the project in their spare time increases risk due to conflicting interests. It is preferable to have dedicated staff working on a project to ensure implementation in a timely manner.

The easiest way to manage the risk is to have slack or contingency periods built into the project plan. This can be used to offset delays and avoid reissuing the project plan. If they are not required, then the project delivers ahead of schedule.

Poor System Performance. This is a classic reason for failure of projects during implementation: the system was sized either by estimation or by a formula. The overall platform performance is not sufficient to operate the system effectively and provide adequate performance to the users and, ultimately, the laboratory customers. Effectively, the system is useless and unable to perform its function. This can be due to a combination of factors:

- Hardware related—processor undersized, insufficient memory, insufficient disc input/output capacity
- Software related—inefficient or non-optimized software routines, database searches slow and not optimized, estimates of laboratory workloads too low

There exist a number of approaches to overcome these problems. One is to define the overall workload of the laboratory accurately and define in unambiguous terms what a sample, test, analysis, and result mean within the context of a specific laboratory. This should allow a vendor to size a system more accurately. Note that, however, vendors work on average system sizes. If a laboratory's application is below average, performance should not be affected and may be enhanced. However, if the application is above average, performance will be affected, often quite dramatically.

Visits to existing users are a more practical way of discovering how effective the vendor has been at sizing a system. If this approach is taken, it is imperative that the site visit is to a laboratory in the same industry and, wherever possible, that it uses the same software modules as the vendor is proposing for you. Often it can be very difficult to find a laboratory site that operates even in the same industry as your laboratory, or if one is found, it is located in a different country. However, a number of aspects of site visits are very useful.

Alternatives exist to avoid performance problems. The approach taken in the author's laboratory was to purchase a small development computer system, develop the software, and carry out performance tests that predicted the size of computer system required to support the intended user base. Another is to carry out a performance test on the potential system configuration, and time the responses obtained. A third is to specify the minimum response times required in the contract with any penalties upon failure to achieve them. The author prefers the more practical approach of direct sizing, as it removes an area of uncertainty during the most critical phase of a project. This practical approach to hardware sizing should also eliminate the need to seek additional funds for a processor upgrade or additional discs soon after the system is operational.

Obsolescence. Given the rapid development and life cycle of hardware and communications components, it will not come as a surprise to find that the system hardware can be replaced in a product line before the organization's depreciation period is completed. If the equipment has been purchased from a recognized supplier, service support should not be a problem but expansion may be. To reduce, but not completely eliminate, this risk, ensure that the development plans of the hardware supplier are known, especially if a proposed hardware system has been available for over two years.

LEARNING FROM FAILURE

Learning from our mistakes is a common saying, but the temptation when faced with a project failure is to keep quiet

or sweep the issue under the carpet. Take another view, as it is unrealistic to anticipate or expect that all automation projects undertaken will be completely successful. The culture of an organization and the attitudes of immediate line management will usually dictate how failure is dealt with: some organizations may encourage risk taking and allow the undertaking of leading-edge projects, with the expectation that some will fail, whilst others may be more circumspect and not encourage risks to be taken. Whatever the organization, it is rare to undertake an investigation of the reasons for failure. This is unfortunate, as failure is a valuable learning experience that should be used and fed back into the cycle of laboratory automation projects for the benefit of the organization.

The causes of automation failures can appear to be many and varied, and failure can also come in varying degrees. However, failure can be classified into four main categories that are presented and discussed below.

1. Failure to Learn

When a project has been undertaken, the lessons and experience should have been learnt and passed to any new project before the latter starts. Knowing the reasons for failure should help a similar project succeed by avoiding the obvious pitfalls.

The corollary, of course, is to know the reasons for a project being successful, which can be just as helpful. Of course, we never bother to understand why a project was successful, do we. Usually it is congratulations and plaudits all around and down to the bar for a pint; however, the dividing line between success and failure can be agonisingly thin.

2. Failure to Anticipate

The essence of a failure to anticipate is not ignorance of the future, as that obviously cannot be foretold, but the failure to take precautions against a known hazard or events. Examples include the rapid development of equipment (scientific, automatic, and computer). It should be possible to anticipate the introduction of new models, usually through vendor briefings under a non-disclosure agreement. Failure to take note of these events could mean the purchase of an item or equipment model that could be obsolete before an automated system is operational. Furthermore, the introduction of any automated system requires careful management of expectation of the potential user base.

3. Failure to Adapt

Adapting can be defined as identifying and taking full advantage of opportunities that arise during the course of an automation project. To exploit opportunities involves having people who have the authority and ability to work independently and use their initiative. Working practices and organizations in a changing environment are not immutable and should alter to meet the new challenges that arise as a result. This needs to be actively managed—never forget that the human element in automation projects is one of the keys for success.

4. Catastrophic Failure

As the title suggests, this is a total failure of an automation project, which can be the result of mistaken scientific principles being applied, the wrong technology being used, non-involvement of the users in the project, or incompetent management. Knowing the general reasons for failure listed above, and the encouragement of a culture of openness and honesty for investigating and explaining failure of individual projects, will benefit all future ones within an organization.

Conclusions

When considering risk assessment and management throughout the lifetime of an automation project, a number of common threads emerge:

- Effective planning is needed, which includes allowances for slippages and tasks that were not identified at the start of the project. The plan should go to a depth that allows the project to progress on strong technical and human grounds. This is not always done, and project plans are usually overoptimistic.
- Communication among all parties (e.g., users, vendor, management, QA, and IT) is an essential element of reducing risk by the transfer of information.
- Discussion of the business benefits of a new system should be realistic to manage user expectations.
- Experience and skills on automation and IT projects are valuable resources within many organizations. Too often they are not used to their full extent by passing experience to different functional groups that are undertaking similar projects. Therefore, many projects waste time and resources overcoming the same problems that other groups resolved on other projects.
- Commonsense and flexible management approaches are essential, both from the user management and from the project manager.
- User involvement is essential for a successful project and must be matched by management commitment.

References

1. McDowall, R. D. Strategic approaches to laboratory automation, chemometrics and intelligent laboratory systems. *Laboratory Information Management* **1992,** *17*, 265–282.
2. McDowall, R. D. The system development life cycle, chemometrics and intelligent laboratory systems. *Laboratory Information Management* **1991,** *13*, 121–133.
3. Royce, W. *Software Project Management - A Unified Framework*; Addison-Wesley: Reading, MA, 1998.
4. McDowall, R. D. The evaluation and management of risk during a laboratory information management system or laboratory automation project, chemometrics and intelligent laboratory systems. *Laboratory Information Management* **1993,** *21*, 1–19.
5. ISO 14971: 2000. *Medical Devices: Application of Risk Management to Medical Devices*; Internal Standards Organisation: Geneva, 2000.
6. McDowall, R. D. Exploiting the benefits of 21 CFR 11. *American Pharmaceutical Review* **2002,** *5*(1).