

Controlling Computers in Business: Selection, Implementation and Testing of Packaged Software

*The fourth in a series of guidance
documents for SMPs and SMEs*

The International Federation of Accountants (IFAC) is the global organization for the accountancy profession with 159 member organizations in 118 countries representing 2.5 million accountants employed in public practice, industry and commerce, government and academe.

IFAC's mission is to serve the public interest, strengthen the worldwide accountancy profession and contribute to the development of strong international economies by establishing and promoting adherence to high-quality professional standards, furthering the international convergence of such standards, and speaking out on public interest issues where the profession's expertise is most relevant.

The IFAC Board established the Small/Medium Practices (SMP) Task Force to investigate ways in which IFAC can respond to the needs of members operating in the small and medium-sized practice whose dealings are principally with small and medium-sized enterprises (SMEs). The SMP Task Force does not issue standards or guidelines such as those set out in the IFAC Handbook. Rather, it is authorized by the IFAC Board to publish the types of documents listed below on issues and practices it considers to be of interest to small and medium-sized practices and enterprises.

1. Guidance documents for small and medium-sized practices and enterprises, which provide practical advice on relevant issues.
2. Research reports, which describe the results of in-depth studies carried out on behalf of the SMP Task Force.

In accordance with these terms of reference, the IFAC SMP Task Force convened two meetings during 2002 for IFAC Member Bodies with particular interests in the SMP/SME area. The principle aim was to ascertain what particular products or services might be of use in the global market place. One such product that met the criteria was a series of guidance documents entitled "Controlling Computers in Business" which was produced under the control of the Information Technology Committee of the Institute of Chartered Accountants of Scotland (ICAS). The majority of the research and drafting in connection with these publications was undertaken by PricewaterhouseCoopers LLP (PwC LLP).

IFAC, in agreement with both ICAS and PwC LLP, has updated the guidance documents under the IFAC banner, with the objective of exposing these documents to the wider SME/SMP market.

This publication is therefore the forth in a series designed to provide practical advice on computing controls. The series, whilst aimed mainly at SMEs, will be of use to SMPs, both for use in their own offices and also for their clients who will mainly be SMEs. Why the Task Force considers computer controls to be of significance to SMPs and SMEs is explained in the Foreword.

The SMP Task Force welcomes any comments you may have. Comments should be sent to:

Technical Manager, SMP Task Force
International Federation of Accountants
545 Fifth Avenue, 14th Floor
New York, NY 10017 USA
Fax: +1 212-286-9344
Email: SMPpubs@ifac.org

Copies of this paper may be downloaded free of charge from the IFAC website at www.ifac.org.

Copyright © December 2003 by the International Federation of Accountants. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the International Federation of Accountants.

This document is summary in form and is not intended to: (i) constitute professional advice or a substitute for professional advice; (ii) be a definitive statement of best practice; (iii) replace the expertise and judgment of your independent accounting, legal, information technology or other professional adviser. Consequently, this document is provided "as-is," with no guarantee of completeness or accuracy, and without warranty of any kind, express or implied, including, but not limited to, warranties of performance, merchantability and fitness for a particular purpose. In no event will PricewaterhouseCoopers, its related partnerships or corporations, or the partners, agents or employees thereof be liable to you or anyone else for any decision made or action taken in reliance on the information in this document or for any consequential, special or similar damages, even if advised of the possibility of such damages.

ISBN: 1-931949-15-8

Foreword

Computers and systems are part of our daily lives. Many of the benefits that were previously derived only from significant investment within large organizations are now available to SMPs and SMEs. Although this brings the potential for substantial business improvement, it also brings the risks associated with ensuring the proper use, management and operational control to deliver the potential returns from systems investment.

SMPs and SMEs need to devote management time to systems management and control issues. To ignore them greatly increases business risk. IFAC's SMP Task Force is aware of the pressures and constraints affecting SMPs and SMEs. The Task Force guidance notes are therefore of a thoroughly practical nature. The Task Force hopes that busy SMP and SME managers will gain considerable benefit from them. The checklists built into the guidance are designed to allow users of the notes to quickly identify if they have any problems. More detailed guidance is provided to assist in the resolution of those problems.

Few organizations do not make use of packaged systems to meet their business needs. This guidance document highlights possible pitfalls that may be encountered in the introduction of package systems and suggests what steps can be taken to ensure a successful implementation.

PwC LLP is heavily involved in organizations within the middle-market and hosts a dedicated website that deals with many of the issues affecting the owners of such enterprises as they try to drive their businesses forward. The address of this website is www.driving-ambition.com, while the main PwC LLP website can be found at www.pwcglobal.com.

Finally, my thanks in particular to Colin Campbell and Victoria Fox of the Glasgow Office of PwC LLP, and to members of the SMP Task Force of IFAC, who are:

Paul Chan, Hong Kong
Ashok Chandak, India
Mohamed Ali Elaouani Cherif, Tunisia
Alex Hilman, Israel
Robin Jarvis, UK
Dawn McGeachy, Canada
Harold Monk, USA
Bernard Scicluna, Malta

ANGELO CASÓ
Chairman, IFAC SMP Task Force

Introduction

1. This is the fourth in a series of guidance notes on computing controls for Small and Medium Practitioners/Enterprises (SMPs/SMEs) that the International Federation of Accountants has produced in association with PricewaterhouseCoopers.
2. Each note discusses an issue relating to computing controls and shows how best practice can be applied to the smaller organization. These notes give information on the issue, including definitions of key terms, costs and benefits, risks and practicalities. Each note then provides a good practice checklist. You should use the checklist to see how well controlled your business's use of computers is against the risks discussed in the note.
3. The readers of these notes will undoubtedly have a wide variety of needs, stemming from two factors. Their level of awareness of the issues discussed will affect how much or how little of each note they will have to use, as will the current level of control in the area discussed. Accordingly, the notes are organized to allow readers to choose the sections they wish to read.
4. Each note has the following sections:
 - Background
 - Key Terms
 - Cost-Benefit Considerations
 - Risk Indicators
 - Practical Considerations
 - Good Practice Checklist and Appendices.

How to Use These Notes

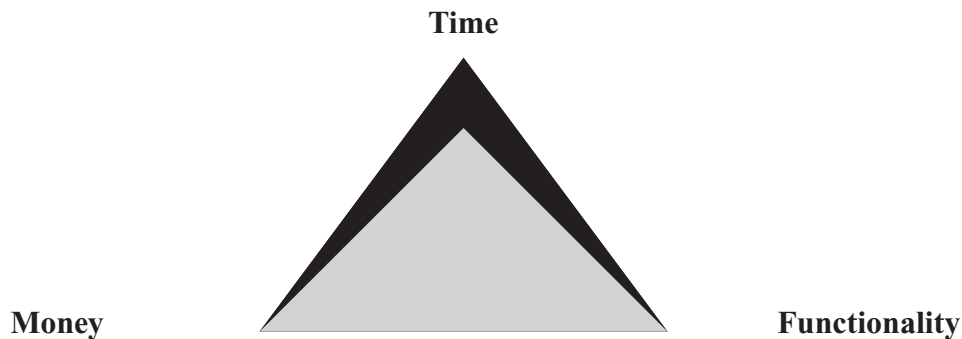
5. If you only have a short amount of time:
 - Read the background.
 - Read as many of the “Key Terms” as you need.
 - Then complete the “Good Practice Checklist,” consider the examples raised in the appendices and complete any appropriate schedules.
6. Read the other sections as required to resolve any issues that might be highlighted by your completion of the checklist.
7. If you have more time, or the checklist results suggest that you need to perform a more detailed review of the issue, consider the other sections on costs and benefits, risks and practical considerations.

Background

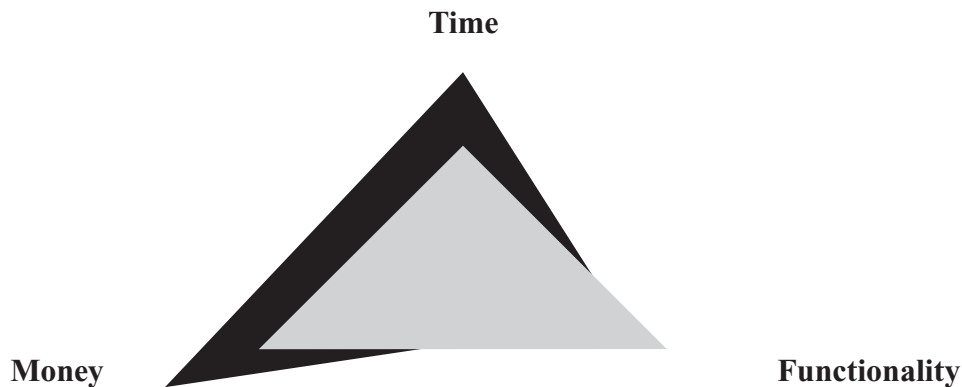
8. One of the most difficult IT-related issues is the implementation of a new computer system. If you are planning for such a change, you must first consider the following:
 - the complexity of your new software;
 - the complexity of your new hardware;
 - migration of your existing legacy systems and data;
 - the potential cultural impact of changes on your business processes;
 - disruption of ongoing business operations as your staff are deployed to support the implementation of the new system;
 - the need to merge disparate groups in your business into one cohesive unit, at least for the duration of the implementation project;
 - the lack of skills and experience with system implementation projects, including appropriate project management skills.
9. All of the above factors individually can create some problems for you, but the implementation of a new computer system often brings them on all at once and can reduce the chance of a successful implementation.
10. Introducing a new computer system into your business may be as simple as implementing new packaged software, developing in-house software or a combination of the two. This guidance note focuses primarily on the implementation of packages — since this is the approach that most SMEs are likely to adopt — and will also touch on the development of “bespoke” (definition in paragraph 22) enhancements offered by package suppliers.
11. The purpose of this guidance note is to alert you to the common pitfalls that such projects might encounter and to give you some practical guidance on the steps to follow to ensure a successful implementation. It cannot, of course, provide a comprehensive project management methodology and, if this is what you need, you may want to obtain professional assistance.
12. Although the focus of this guidance note is on the selection and implementation of a new financial system, the principles may apply equally to non-financial systems.
13. There are three main factors involved in implementing a new computer system:
 - the elapsed time necessary to introduce the new system;
 - the resources, including money, necessary to procure the system and implementation support; and
 - the functionality delivered by the new system.
14. Elapsed time is a consideration since you are likely introducing the new computer application to meet a business need and expect that need to be met within a reasonable

timeframe. You may want the new system in place to deal with the introduction of a new regulation or you may simply want it to cope with a new financial year.

15. Resources generally mean money but can also include the need for key staff at critical points in a project. For simplicity, this guidance note uses the term money. Money incorporates the cost of buying the software and hardware, and any programming required, but also includes the user cost of implementation.
16. Functionality represents the business functions the new system can provide and might include the automation of existing manual tasks or the provision of functionality that is not present in your current business processes.
17. The definition of a successful implementation project is one that delivers the required functionality on time and within budget. The balancing of these three variables is the ultimate aim of the implementation project manager, and the importance attached to each of the three will prompt you to make different decisions about the management of the project.
18. The following diagram shows a project where the necessary functionality was delivered on budget but the elapsed time was greater than anticipated. The gray triangle represents the initially planned outcome, while the black triangle represents what actually happened:



19. Most projects do not come in on budget or on time, and a significant number do not deliver the functionality anticipated at the outset. The following diagram shows a typical project's outcome:



20. You might find these graphs useful in managing your project. If money is your ultimate constraint, you can either reduce the delivered functionality or extend the delivery deadline to deal with that constraint. If time is of the essence, you can reduce functionality and make additional resources available to meet the deadline. Finally, if functionality is the most important factor, you can extend the implementation date and/or allocate additional resources to the project.
21. When you decide to implement a new computer system, it is important to be very clear at the outset why you think you need that system, what benefits it is to deliver and when those benefits must be delivered.

Key Terms

22. **Bespoke** A piece of software written specifically for a customer rather than being part of any particular package. Bespoke software may be used to supplement the functionality of a particular package or to deliver entire business applications.
23. **Interface** An automated procedure that allows information in one computer system to be passed to another computer system for processing. An example of an interface might be between a bespoke sales order processing system and a sales ledger package. Order details would be captured and priced within the sales order processing system and then passed to the sales ledger to produce the physical invoice and update the customer's ledger account. In due course, the sales ledger would interface with the general ledger to update the necessary account details.

Cost-Benefit Considerations

24. Since the selection and implementation of a new computer system is one of the most significant decisions you can make, you really ought to do a formal cost-benefit analysis for all but the most insignificant implementations. Such an analysis will point the way to your best solution as well as justify your eventual choice.
25. Be sure to include the following types of costs in such an analysis:
 - hardware;
 - communications infrastructure;
 - software;
 - training;
 - implementation assistance;
 - stationery;

- alternative computer disaster recovery arrangements;
 - hardware support;
 - software support;
 - software enhancements;
 - operating system enhancements;
 - internal project management;
 - internal information systems staff effort.
26. The financial benefits might include:
- reduced staffing levels;
 - reduced administrative costs;
 - reduced hardware support costs;
 - reduced software support costs;
 - increased turnover;
 - reduced finance charges.
27. Other qualitative benefits that are more difficult to value include:
- improved management information;
 - higher barrier costs for entry into your particular market;
 - improved customer service;
 - fewer complaints;
 - reduction in errors.
28. A potential framework for documenting a cost-benefit analysis is included as Appendix I in this guidance note.

Risk Indicators

29. If you are implementing a new computer system, you need to consider the following risk factors:
- the urgency of your business need;
 - what functionality you really require;
 - the financial stability of the package supplier;
 - the frequency of software updates;
 - the stability of the hardware platform;

- industry acceptance of the individual system components;
 - the number of other users of that software.
30. Risks can be placed into three main categories:
- project risks;
 - technology risks; and
 - financial risks
31. You need to manage each of these risk categories to ensure a successful implementation. Maintain a risk log throughout the project and allocate each risk to a responsible person, who will be considered the owner of that risk. The role of each risk owner is to take responsibility, throughout the duration of the project, for monitoring the potential likelihood of the risk becoming a reality, what the impact of that would be and to implement appropriate countermeasures.
32. Project risks are the ones associated with the management and control of the project itself. These can include such factors as:
- level of senior management commitment;
 - experience and ability of project management staff;
 - natural tendency to underestimate project effort or elapsed time required;
 - availability of project team staff.
33. Technology risk stems from using the proposed technology to support key business processes. For example, you may find it acceptable to base a small bespoke database development on a new hardware platform because the database is not critical to your operations. You would not, however, contemplate using that same hardware for your mission critical computer systems because of its untried nature and the lack of a track record.
34. The use of pilot systems is one way of reducing technology risk. The level of risk you might be willing to take will depend on whether you see your business as a technology leader and the extent of competitive advantage that the new technology might deliver.
35. Financial risks are the uncertainties relating to project estimates. The estimates include budgets for any bespoke tailoring of packaged software and implementation assistance from third parties. Many projects come in over budget solely because the initial estimates were too optimistic. Inexperienced project managers neglect to take into account the additional costs that increased elapsed time will incur. The development of a piece of interface code might take the time originally anticipated but, if it is delivered two weeks late, additional costs will be incurred while waiting for that delivery.

Practical Considerations

36. All systems implementations are different. Although two businesses may be implementing exactly the same package, the businesses themselves will be different. They may have different links with the outside world, different perceptions of their information needs and may have allocated different priorities to their implementation projects.
37. Systems implementations are as much, if not more, about the human dimension involved in systems change. Very rarely will a new system be implemented with no changes to the manual processes and procedures that surround that system. Therefore, at the outset of any implementation, it is essential that you recognize and incorporate that human dimension.

Definition of Requirements

38. When you are planning to implement a new computer system, it is extremely important to begin with a formal definition of your requirements. Although, for smaller implementations, the preparation of a formally approved document may seem onerous, it is still recommended because:
 - it documents what functionality you expect the new system to deliver;
 - it records management approval of that functionality;
 - it builds consensus among the users;
 - it defines the scope of the information systems to be implemented.
39. A pitfall to be aware of throughout any systems implementation is the phenomenon known as “creeping scope.” Users may agree to an initial specification of requirements but, afterwards, want the system to deliver more and more functionality.
40. Some specification, selection and implementation projects, even for smaller and medium-sized enterprises, can last between one and two years. Meanwhile, technology changes, improves and is enhanced. Inevitably, users' perceptions, expectations and requirements will tend to grow dynamically. Because this creeping scope will almost certainly increase the time and money necessary to deliver your system, you must manage the process very carefully. There may be occasions when increased functionality will deliver a real business benefit and you should include it within the scope of your project. In that case, be sure to explicitly identify the increased time and money requirements so that user and management expectations are properly managed.
41. To ensure that the definition of requirements is properly carried out, you need to put together a project team. Ideally, the team should be led by a senior member of a user department and should comprise a mixture of users and IT staff. It is important that users take responsibility for the system at the outset since it is their system that is being implemented. If you have no in-house IT expertise, you must obtain an appropriate level of quality support from an external supplier.

42. All the members of the project team should have their normal duties reduced so that they can devote the necessary time to the project. At certain times, the team will probably be working full-time on the project.
43. The project should be sponsored by a senior member of your management team. This sponsor should have the time to receive and act on regular progress reports.
44. When documenting the necessary requirements for a new computer system, you need to look at the following:
 - functionality;
 - reporting; and
 - security.
45. The requirements specification should contain, in non-technical language, a summary of what you want the new computer system to do. It may be tempting to avoid preparing such a requirements specification in the belief that your particular company's requirements are totally generic and can be met by standard software packages. There is, however, a danger in assuming too much and finding that the package that is eventually selected either does not do everything you need or that it requires extensive and expensive customization.
46. Sometimes, it may be cheaper and more efficient to alter some of your existing business processes rather than procure a package that is an exact fit. It is better to explicitly identify such processes rather than to find them after you have begun the implementation.
47. In any specification and selection exercise, you want to link your new system directly to your business strategy. The linking of information systems strategy to business strategy is important to ensure that your management has access to the quantitative and qualitative measures required to make decisions quickly and accurately.
48. When drafting a requirements specification, bear in mind the following factors:
 - What are the critical success factors for your business, i.e., what things, if done well, will contribute to the success of your business?
 - What business processes are being computerized?
 - What requirements must the system meet, e.g., the ability to process sales orders on-line?
 - What volume of transactions is to be processed?
 - What is the frequency of processing, e.g., ad-hoc, daily, monthly?
 - How desirable are each of your requirements, e.g., mandatory, highly desirable, desirable?
49. The use of a standard form will allow you to rank different packages. You can then score each package as to how it meets the necessary requirements. When scoring different package attributes, use weighting factors to attribute different levels of importance to

different selection criteria. For example, the financial stability of the package supplier should carry more weight than the ability to change screen colors.

Functionality

50. Functionality refers to the processing steps that that the computer system is to carry out. For a sales order processing system, for example, the necessary functions might include the ability to:
 - process orders on-line;
 - add new customers during the order-taking process;
 - enter different measures of quantity for the same product, e.g., pallets, boxes, individual units;
 - automatically generate a sequential sales order number;
 - enter a number of order lines for one order number;
 - automatically generate an order confirmation;
 - enter a different delivery address for that particular order.
51. Generic application templates are available that you can use to ease the process of defining what functionality you require.
52. The important aspect of defining functionality relates to areas where your business may have specific requirements. For example, a steel stockholder will require a stock control system that can accommodate flexible stock handling. Within this business, a piece of stock may be cut and the majority of it sold, but the remaining pieces become another type of stock. This type of functionality is not one you would normally find in a standard manufacturing stock system.
53. Other requirements that not all packages might be able to provide include:
 - multi-currency;
 - the ability to deal in more than one base currency;
 - multi-company, with consolidation;
 - multi-lingual;
 - the ability to meet the fiscal reporting requirements of different countries;
 - commitment accounting;
 - links to PC spreadsheet or database packages;
 - links to suppliers and customers systems utilizing EDI, XML and .NET technologies.
54. Each company needs to examine its core business processes and identify all such requirements because it is often these types of requirement that will restrict the available choice of packages.

55. Although packaged software tends to provide similar financial modules, such as general ledger, sales ledger and purchase ledger, different packages may provide different functionality. For example, a sales ledger package may or may not include sales order processing. Be sure to assess the price for two different solutions based on the delivery of functionality, irrespective of the number of modules that you may have to purchase.
56. Another factor to consider is that some packages use a baseline module to provide functionality, such as system administration, security and control of interfaces to other systems. This type of module provides no direct business functionality but is required to operate the software.

Reporting

57. Packages very often come with report writing facilities that can interrogate the underlying data files. Some of these facilities are difficult to use and it is worthwhile ensuring that all your basic reporting requirements are met by the standard package — or that the supplier will write the reports and provide them as part of the implementation service.
58. Users' initial views of a new system will be strongly influenced by their ability to extract key reports. Sadly, it is often the availability of such key reports that is sacrificed in the final stages of an implementation.

Security

59. As mentioned in earlier guidance notes in this series, adequate computer security depends on a number of different security mechanisms at the physical, operating system and application system level. As a minimum, any computer application should provide the ability to secure access to sensitive facilities, such as security administration, and should also be able to segregate incompatible business functions.
60. A differentiator of packages is the quality of their security reporting. Most packages will allow you to produce a report showing what facilities an individual can access. Say, for example, that John Smith can access the system to set up a new supplier and process purchase orders. But very few packages produce reports showing which individuals have access to a specific facility, for example, the fact that only Fred Jones and Sam Weaver have access to security administration.
61. Passwords should be encrypted so that other system users cannot discover them; ideally, the User ID should be capable of being linked to the operating system password. If such a link is not possible, you need to consider whether the application password should be aged so that it expires.

Selection of Solution

62. Once you have defined your requirements, the next step is to identify a number of packages that you believe might meet your requirements. Generally, the packaged software

market is divided up into different segments, with package suppliers aiming at different sizes of client. There is also a wide range of solutions, covering different industry sectors and organization size.

63. Virtually all accountancy and software magazines contain advertisements for software suppliers. Phone calls to these software suppliers will be met with a deluge of information that you can filter to select a short list of three or four potential candidates.
64. You should then issue a formal Invitation to Tender (ITT) to the selected number of software suppliers that have packages apparently meeting your requirements specification. The ITT will ask each supplier to state whether its package meets all of your required functionality, how much bespoke development work will be needed to deliver that functionality and for a quote on the product.
65. Attend supplier presentations but ensure that you give them a script to follow in advance. If there are particular aspects of their package you are interested in, be sure to give them sufficient time to cover those areas. All suppliers should also be given sufficient free time to demonstrate how their products differ from the competition. Consider providing suppliers with examples of the types of reports you already use and ask them to replicate such reports with their own package, perhaps using the report generation facilities that their package includes.
66. Have the entire project team attend all supplier presentations and allow sufficient time for team feedback and discussion at the conclusion of each presentation.
67. Reference sites are another useful mechanism for assessing the validity of suppliers' representations. Ask all software suppliers for a list of customers that operate in your industry and select a random sample to visit. Arrange to visit these sites without the supplier and obtain an unbiased assessment of the package and the supplier's support.
68. As well as the fit with the functionality you need, other factors may also rightly influence your decision as to which software package to select. These can include:
 - Supplier culture — How does the culture of the software supplier fit with yours? Is the supplier aggressive, helpful?
 - How restrictive is the supplier's contract?
 - Implementation support — Has the supplier convinced you that it has the resources to assist you with your implementation and are you an important customer for that supplier?
 - User base — The number of other users of that particular package will provide an indication of how stable the product is but may also adversely affect the level of support available from the supplier. How many users are within your industry sector?
 - Ongoing software support — Although software suppliers generally provide some implementation support as part of their standard package, you need to know about the quality and quantity of support that will be provided on an ongoing basis. Don't

overplay this concern because most businesses don't require a significant amount of support anyway. It may be cheaper to negotiate a lower initial package cost and pay for subsequent support rather than to take a more expensive all-inclusive package initially.

- Training — How much training is provided as part of implementation and for how long a period will that training be provided?
 - Future upgrades — Will future package upgrades be provided as part of the ongoing software support contract or do they have to be paid for? How often are upgrades issued and how much will they cost? Be sure to consider the hardware implications of upgrading to later software versions.
 - Supplier viability — Does the supplier have a successful track record and is it still likely to be around in three years time?
 - Is a secure test environment supported so that new releases can be installed and tested without compromising the existing live environment?
69. Although you should be implementing new computer systems based on the best fit for your business, you should also give some consideration to the hardware on which the system runs. For example, the best package may run on an insecure, esoteric or obsolete hardware platform. An alternative option, therefore, may be to select a package that delivers a less perfect business fit but that runs on a more mainstream hardware platform or on one you already have some experience with.
70. When considering the relative benefits of different packages, be sure to quantify the total costs of the different solutions including:
- hardware;
 - software;
 - training — in both hardware and software;
 - implementation support;
 - business process re-engineering; and
 - consulting.
71. Many software packages rely on a database management system that supports the program code and provides access to data. When selecting a package, you must check out the hidden cost of maintaining this database management system. Some suppliers will provide database support as part of their overall package support, while others require you to take out separate license and support agreements with the database management supplier. Such additional contracts can significantly increase the overall cost of a package. Similar problems can exist with third party report writers that may initially be bundled with a package.
72. Another factor when appraising packages is where on the technology curve you want to be. Although leading-edge companies will take greater risks with their use of information technology, they may also reap greater benefits. IT staff may prefer to select the leading-edge option because they will see it as a way of enhancing their own CV, which will make

them more marketable. After a package implementation, software suppliers occasionally recruit good client staff. You should ensure that the contract with your software supplier prohibits such poaching.

73. You can gather valuable information about package suppliers and their products from their user groups. Aim to attend some user group meetings and assess the level of satisfaction expressed by existing package users. Also, consider the relationship between a package supplier and its user group. Suppliers often use the existence of a user group as a selling feature but then do not support the group in practice.
74. Ask to see copies of the documentation provided for the software. As a minimum, there should be technical systems documentation that allows in-house IT staff to understand the technical details behind the operation of the software. There should also be user documentation that tells users how to exploit the features of the software. Note that this user documentation is not a substitute for the development of your own user procedures, since the package supplier will document only how the software operates. Your own user procedures should also include the manual processes and controls around the software itself.
75. Confirm quoted pricing in detail and ask how the supplier has arrived at that pricing. Determine what ensures that a supplier's pricing commitments will be met. Also test the reality of the quoted expected time to implement.
76. When selecting your ideal solution, carefully review the supplier's contractual terms. Ensure that no payment is due until the system has been satisfactorily implemented and all bespoke enhancements have been delivered. Also, ensure that penalty clauses are incorporated as a disincentive for late software deliveries or poor quality software releases.
77. Escrow agreements provide for a copy of the source code of the software to be deposited with a third party so that, should a package supplier go out of business, you can still gain access to the underlying program code.
78. When reviewing the terms of a supplier's escrow agreement, consider:
 - how often the source code is updated;
 - the current version of software already held in escrow for other customers;
 - the availability of program documentation;
 - the availability of a compiler to generate object code from the underlying source code.

Implementation of Solution

Testing

79. Once you have selected and acquired your new system, you have to test it. The following types of testing may be carried out:

- | | |
|-------------------------|---|
| System testing | The purpose of system testing is to check that the system operates as it is intended to. In a package implementation environment, the software supplier should be doing the system testing and you should obtain evidence from the supplier on the tests carried out and the results of those tests. |
| User acceptance testing | UAT is carried out by potential users of the system to ensure that they are happy with the operation of the new software. Questions will normally focus on the look and feel of the system and how output is delivered. UAT also focuses on the manual processes and controls around the new system to ensure that the entire business process can operate in an efficient and controlled manner. |
| Volume testing | The system needs to be tested to ensure that it can handle the volume of transactions that your business needs to process. The software supplier will have made representations about the volume of transactions that the software can process, but these statements can be idealistic and volume testing seeks to confirm that the system capacity is adequate. |
| Stress testing | Stress testing is a specialized form of volume testing. This is where you attempt to create an unrealistic workload for the system to assess how the system will react. Stress testing can be achieved by having a large number of on-line users processing complex transactions while, at the same time, submitting reports for processing and posing complex enquiries of the system. |
| Integration testing | Integration testing is performed to ensure that all the component parts of the system operate together and with any other software that your business has interfaced with the new system. |
| Regression testing | Should the new system fail, it is important that you can revert to your previous method of processing. Regression testing verifies that information from the new system can be input into the old system and that adequate procedures have been put in place to deal with this eventuality. |
80. All testing should be performed on the hardware platform on which the final package is to operate.
81. Your first step in testing a system should be to prepare a test strategy that records what types of test should be carried out. For example, the volume of transactions expected to be processed may be so low that volume testing of one particular system may not be appropriate. The reason for not carrying out a particular test should, however, be recorded so that the assumptions underlying that decision can be scrutinized later.

82. Each type of test should ideally have a detailed test plan that records the main objectives of testing that particular system. For example, the user acceptance test plan for a sales order processing system might state that the purpose of the test program is to ensure that:
- a) the sales order processing system operates correctly for all types of transactions processed;
 - b) the necessary information is correctly passed to other related systems;
 - c) reporting from the system is accurate;
 - d) manual procedures around the system are practical and implemented an adequate level of control.
83. For each of these objectives, there will be a detailed test that records the steps to accomplish the overall objectives. For example, under the second objective noted above, information would be traced from the sales order processing system to the sales ledger. The detailed test should define exactly what has to be done, what the anticipated result of that test will be and what has to happen should the test fail. All tests that fail at the detailed stage must be investigated and the problem resolved. The particular test steps should then be re-performed to ensure that the problem was resolved satisfactorily.
84. During testing, be sure to allocate the responsibility for establishing test data, generating the expected results and verifying output to knowledgeable staff. Many system problems identified after live implementation should actually have been picked up during testing. The common reasons why testing fails are:
- The test plan does not cover all eventualities.
 - No one verifies that all tests were performed satisfactorily.
 - Users find a problem during testing but don't send it to the appropriate person for resolution.
 - Problems that are sent to management are not investigated and resolved before implementation.
 - User procedures are not tested as part of system testing.

Implementation Strategies

85. New systems can be implemented in a number of different ways:
- “big bang” — where the new system immediately takes over the functionality of the old system at one particular point in time;
 - phased — where the new system takes over functionality in stages, e.g., implementing the stock control module first;
 - parallel running — where the new and old systems run together for a period of time and results are compared.

86. The advantages and disadvantages of each of these strategies are discussed in the table below:

| | Advantage | Disadvantage |
|------------------|---|--|
| Big Bang | No lengthy implementation period | Risky as the consequences of failure could be significant Requires exhaustive testing of the new system before it should be considered as a viable option |
| Phased | The new system is implemented in manageable pieces | Interface problems between old and new systems — how do you update the existing general ledger with transactions recorded in the new stock control system? The implementation period can be quite lengthy |
| Parallel running | The new system does not go fully live until it is proven in the company's own environment | Expensive to maintain two systems for any length of time Staff can be put under significant pressure maintaining two systems and reconciling the results |

87. You will decide on what the appropriate strategy for you is depending on how important the new system is to your business, the confidence in the system after testing has been completed and the availability of implementation resources.

Timing of the Implementation

88. When implementing financial systems, you need to determine the best time to implement them. Both financial reporting and regulatory timetables will influence when a new system should go live.

89. Organizations often attempt to implement a new system at the beginning of a new financial year so that it will deal with all of the new year's transactions. This would mean, however, that users would be testing the system at the same time as they are preparing for year-end reporting requirements. Any problems with the new system will occur at the same time as the financial accounts for the previous year are being prepared.
90. If your industry is regulated, it may be appropriate to ensure that all the regulatory returns for a particular period can be produced from the one system.
91. It may be better to consider implementing a new system mid-way through a financial period, thus reducing the stress on financial staff. Arrangements will, however, need to be made to transfer current year balances for the nominal ledger and to consider whether management's reporting requirements could be met in this situation.
92. When you decide to implement your new system, it will be necessary to consider how best to deal with debtor and creditor balances. It might be better to continue to use your old system to run down debtor and creditor balances until the remaining amounts can be tidied up, or you may wish to use the new system to manage those amounts. The cost of software and hardware support for the old system will normally determine how quickly you want to get rid of the old system.
93. Other considerations when planning how long to retain an old system include the need to retain historical data in a readily accessible form. Do you run the old system for a period or transfer transactions across? Are there record retention rules that require information to be kept on-line for access? It may often be more expensive to transfer historical data than just to keep the old system running.
94. When starting a package selection and implementation project, it is important to consider the impact the new package might have on existing manual procedures and controls. It is also important to consider whether the introduction of the new system can be used to streamline existing processes though some process change.
95. To take advantage of any changes in manual procedures and controls, be sure to involve a sufficient number of users in the implementation plans so that your user community will accept the new practices. A number of unsuccessful implementations have occurred solely because users were not adequately informed of procedure changes and did not appreciate the reasons for the changes. Consequently, the software was not used in the way it was intended.
96. Almost all your staff will require training on the new system. User procedures need to be documented at an appropriate level so that your users appreciate what they are doing and why they are doing it. A package supplier will often offer a number of days' training and you should critically review how much you think you will need. Surprisingly, most organizations do not use the full amount of training a supplier provides and, instead, ask for a larger proportion of the necessary support from in-house "experts" who quickly exploit the features of the new system.

97. All staff involved with the implementation of a new system will obtain valuable experience with the particular hardware and software solution you have selected. It is important that this experience be shared throughout the team and that knowledge not be restricted to one individual.

Post-Implementation Support

98. After the system has gone live, there is still a bedding-down period where post-implementation support must be provided.
99. The focus immediately after implementation must be on the identification and resolution of problems, no matter how insignificant those problems might appear. Users may notice that things are not correct within the new system but may feel intimidated about questioning why certain things are happening. Encourage your users to ask questions because, often, they are the first ones to spot when processing is going awry.
100. After implementation, direct some specific attention to the first period-end process to ensure that all reconciliations are properly carried out and that no differences have occurred. Again, it is possible that something has not been set up correctly within the system and it is only the manual period-end reconciliation that will identify that problem.

Good Practice Checklist

| | Yes | No | N/A | Reference |
|---|-----|----|-----|-----------|
| A. DEFINITION OF REQUIREMENTS | | | | |
| Establishing a project team | | | | |
| Have you set up a project team to select and implement your new system? | | | | |
| Does a senior member of a relevant user department head the team? | | | | |
| Does the team comprise a mixture of users and IT staff? | | | | |
| Have you arranged to reduce the normal workloads of all staff who are part of the project team? | | | | |
| Justifying the decision to implement new systems | | | | |
| Have you done a cost-benefit analysis that justifies the decision to replace the existing systems? | | | | |
| Has the cost-benefit analysis included costs for: hardware, including desktop computers communications infrastructure software training implementation assistance stationery alternative computer disaster recovery arrangements hardware support software support software enhancements operating system enhancements? | | | | |
| Have you considered “do nothing” as a viable option? | | | | |
| Documenting requirements | | | | |
| Have you prepared a formal requirements specification? | | | | |

| | Yes | No | N/A | Reference |
|--|-----|----|-----|-----------|
| Does the specification contain details about the following requirement areas: functionality reporting security? | | | | |
| Have all interested users and IT staff read it and agreed to it? | | | | |
| Have you ensured that all changes to the original requirements are documented and that implications for elapsed time and increased cost are explicitly identified? | | | | |
| Have you identified and documented any necessary interfaces to existing systems? | | | | |
| Have you mapped your requirements against a standard application template for that type of system as a check for completeness? | | | | |
| Have you paid adequate attention to the parts of your company's processes that are unique to your industry? | | | | |
| Do your requirements document the functionality you need rather than specify the type of software you think you need? For example, you need a mechanism for processing sales orders rather than a sales order processing module. | | | | |
| B. SELECTION OF SOLUTION | | | | |
| Have you and your management assessed available packages and identified a manageable number of potential solutions? | | | | |
| Have you distributed the requirements specification to a short list of suppliers and mapped their offerings against it? | | | | |
| Have you scored the supplier submissions to identify which package is the best fit for your business, bearing in mind the distinction between must have, desirable and highly desirable? | | | | |
| Have you prepared a demonstration script for each supplier documenting what particular aspects of functionality you want covered during the demonstration? | | | | |
| Have you allowed enough time for the supplier to demonstrate other aspects of the system? | | | | |

| | Yes | No | N/A | Reference |
|--|-----|----|-----|-----------|
| Did the supplier give you a list of all customers of the package so you can choose a selection of reference sites (rather than selecting its best customers)? | | | | |
| Have you contacted and visited the reference sites? | | | | |
| Is the supplier financially stable? | | | | |
| Does the package supplier have sufficient support staff to assist with implementation? | | | | |
| Does the quote for the package include the right amount of training for your particular needs? | | | | |
| Will the package supplier enter into an escrow agreement for the provision of source code? | | | | |
| Does the escrow agreement provide for: frequent updates the provision of program documentation the provision of a suitable compiler? | | | | |
| C. IMPLEMENTATION OF SOLUTION | | | | |
| Have you chosen a suitable date for implementation? | | | | |
| Has adequate testing been carried out? Consider: system testing user acceptance testing volume testing stress testing integration testing regression testing | | | | |
| Have user procedures been prepared and have senior members of your user departments agreed to them? | | | | |
| Have you kept your staff informed about the progress on the project? | | | | |
| Has your staff received adequate training, not just on the computerized aspects of the system, but also on the pertinent manual procedures? | | | | |

| | Yes | No | N/A | Reference |
|---|-----|----|-----|-----------|
| D. POST-IMPLEMENTATION SUPPORT | | | | |
| Have procedures been developed to capture and act on any issues that might arise after the system is implemented? | | | | |
| Have users been educated on the need to be vigilant and that “no problem is too small”? | | | | |
| Have you asked the supplier for appropriate support procedures? | | | | |

APPENDIX I

POTENTIAL FRAMEWORK FOR A COST-BENEFIT ANALYSIS

| | | | | | |
|------------------------------|----------|---------------|----------|----------|----------|
| Option no. | | Option title: | | | |
| Option description: | | | | | |
| Benefits of option | Period 1 | Period 2 | Period 3 | Period 4 | Period 5 |
| (all figures \$000s) | | | | | |
| Total benefits | | | | | |
| Costs of option | | | | | |
| | | | | | |
| Total costs | | | | | |
| Net benefits (costs) | | | | | |
| Net present value at 5% | | | | | |
| Cumulative benefits (costs) | | | | | |
| Cumulative net present value | | | | | |

APPENDIX II

EXAMPLES OF IMPLEMENTATION FAILURES

Improper specification of requirements

A consortium with a number of different companies in different locations wanted new software to handle its financial and manufacturing systems. It selected a package that provided all the necessary functionality for a single company. Unfortunately, it did not fully consider the size of the entire consortium and was misled by supplier claims as to the software's capabilities in that area.

The consortium purchased the system and soon found that it had to have multiple copies of the software running on one machine to enable each of the individual companies to process its own information within its own database. This affected the capacity of the hardware required to run the new system and increased the implementation and support costs.

No regard for technology

Another company chose a package that the software vendor stated could run on a number of different types of computers. The customer opted for one specific hardware vendor and, part way through the new system's implementation, discovered that the software vendor had never before implemented its software on that particular hardware.

The customer eventually provided the package supplier with time on its own computer so that the supplier could test its code before the software package was delivered. Costs increased because of this situation and support was more difficult to obtain because the package vendor tried to convince the company that every software problem encountered could be attributed to the different hardware platform being used.

Improper control of suppliers

A business purchased a package from a vendor and also paid for bespoke development work to allow the package to interface with its existing systems. The customer paid the supplier for both the package and the bespoke development work before receiving delivery and before being satisfied that the system operated satisfactorily.

Unbeknown to the customer, the package vendor had also sold the package with some bespoke tailoring to another organization that refused to pay for the software until it was operating correctly. As a consequence, all of the vendor's efforts went to satisfy the second customer and no efforts were directed toward the organization that had paid up front.

Inadequate contract negotiation

A business selected a packaged solution from a software vendor because it met the majority of its business needs and was technologically very advanced. During the contract negotiation phase, however, the customer failed to insist on penalty clauses for non-delivery of bespoke pieces of software, even though, during the selection phase, it had recognized that support was an area of concern.

During implementation, the bespoke pieces of software were delivered late, had not been tested adequately and often introduced bugs into the standard package. Unfortunately, the customer had no sanction against the supplier and was at its mercy until the code was finally delivered.

Poor supplier support

A company selected a well-known accountancy package and employed a recognized consultant, recommended by the software supplier, to assist with the configuration and implementation of the software.

During the implementation, it became apparent that the consultant had only recently been trained in the use of the software and did not have enough business knowledge to add much value to the configuration process. The software supplier needed more implementation consultants and had chosen this particular implementation to train one of its consultants.

Failure to properly investigate reference sites

A bank in Eastern Europe wished to implement an integrated banking system and was approached by a large hardware and software vendor with a package that apparently satisfied the bank's needs. The supplier indicated that a large UK Building Society had already implemented the software successfully. Based on that claim, the bank purchased the software and then encountered significant implementation problems.

It later transpired that the UK Building Society had been given the name of the East European bank as a previous customer of that same banking package and had purchased the software on the strength of that recommendation. Neither party contacted the other reference site and both had significant implementation difficulties.



International Federation of Accountants

545 Fifth Avenue, 14th Floor, New York, NY 10017 USA

Tel +1 (212) 286-9344 Fax +1 (212) 286-9570 www.ifac.org