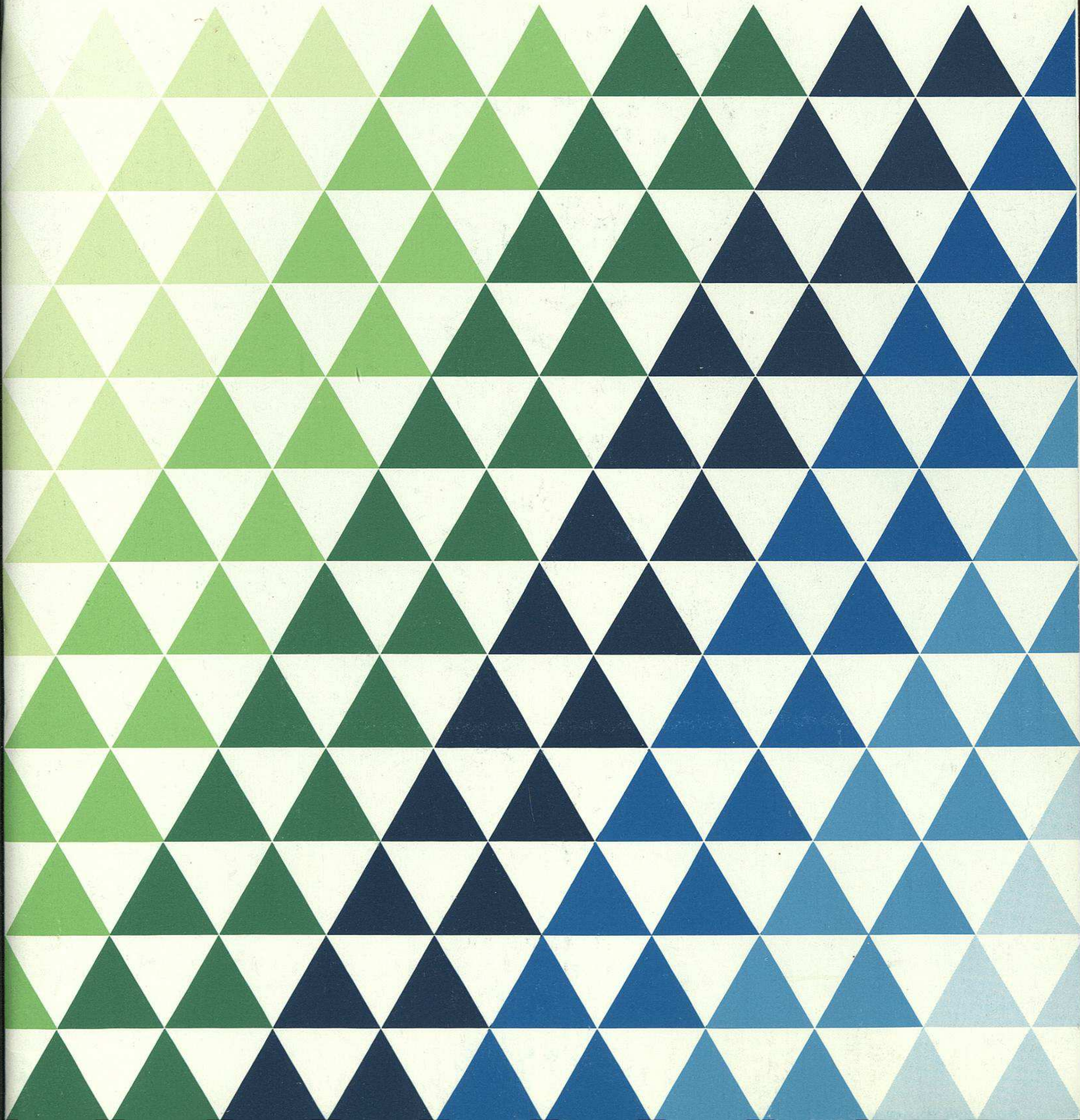
Management Summary

BUTLER COX
FOUNDATION

Using System Development Methods



BUTLER COX
FOUNDATION

Using System Development Methods

Management Summary
Research Report 57, June 1987

Butler Cox & Partners Limited

LONDON
AMSTERDAM MUNICH NEW YORK PARIS

Published by Butler Cox & Partners Limited
Butler Cox House
12 Bloomsbury Square
London WC1A 2LL
England

Availability of reports
Members of the Butler Cox Foundation receive three copies of each report upon publication; additional copies and copies of earlier reports may be purchased by members from Butler Cox.

Printed in Great Britain by Flexiprint Ltd., Lancing Sussex.

Using System Development Methods

Management Summary

This document is the Management Summary of Foundation Research Report 57, published in June 1987. The full report is available to members of the Butler Cox Foundation.

In the continuing struggle to improve the quality, cost, and timeliness of information systems, many organisations have invested heavily in proprietary system development methods, techniques, and tools.

To understand system development methods and their purpose it is necessary to cast some light on the confusion of terms currently in use. We formally define the terms – techniques, methods, and tools – in the main report, but basically a *technique* (data analysis, for example) is the procedure for carrying out a particular development activity; a *method* is a way of using a technique (or a combination of techniques) for one or more development phases; and a *tool* is an automated aid (usually software) to help in using a technique or method.

Costs are high, but benefits can be substantial

The total cost of introducing proprietary methods can be high – running to hundreds of thousands, or even millions, of dollars by the time allowance is made for the cost of the methods and their supporting tools, the cost of training staff to use the methods, and the 'lost-opportunity' costs incurred whilst the methods are being implemented. Nevertheless, the benefits of using system development methods can be substantial and they fall into two main categories: Improved quality of systems; and improved control over the development process itself in terms of both cost and time. Development methods provide these benefits because they can:

- Promote a better fit between the business requirement and the end-product by ensuring that the definitional and analysis phases are thorough and complete, and by improving communications between developers and users.

- Ensure that proper objectives, milestones, and budgets exist, and they provide the means to measure progress against them objectively.

- Avoid effort being wasted through confused responsibilities, duplication of work, working at the wrong level of detail, or developing conflicting in-house practices.

- Encourage better utilisation of staff by promoting conformity of working practice and, hence, mobility across projects.

However, introducing a method does not of itself necessarily improve development productivity. Productivity gains (developing systems less expensively and with less effort) come from using tools that support methods.

No single method provides a complete solution

The demands on a typical systems department are so diverse that several different development processes may need to be used at any one time. Unfortunately, no one of the current proprietary methods covers the range of different processes. (The process most frequently used is the conventional linear process, which uses a staged development cycle and proceeds linearly through analysis, design, construction, testing, and implementation; other development processes include iterative, small-system, application-package, and accelerated. Their characteristics are described in the main report.) We analysed a sample of 40 proprietary methods and tools and found that none of them covers all of the activities of even the conventional linear process (see Figures 1 and 2 overleaf). Moreover, none of them could be used for all types of development process. For example:

- Some methods (MCP and Method 1 are two examples) concentrate on project management and on defining what needs to be done (objectives, tasks, and deliverables of each phase). They tend to be bureaucratic and documentation-oriented and, as such, are inappropriate for small developments and nonspecialist users. At the same time, they give little assistance on how to do it' – for example, no detailed estimating help.

- Other methods focus only on a single phase or even part of a phase of the development cycle –

Figure 1 No one type of method completely covers every phase of conventional development

	Type of method									
					Multiphase					
	Management		*Single-phase*		*Analysis and design*		*System-build*		*Integrated*	
Development phase	*How*	*What*	*How*	*What*	*How*	*What*	*How*	*What*	*How*	*What*
Survey and feasibility		▨	▨	▨					▨	▨
Requirements analysis		▨	▨	▨					▨	▨
Systems analysis		▨			▨	▨			▨	▨
Systems design		▨			▨	▨			▨	▨
Programming		▨					▨	▨	▨	▨
Testing		▨					▨	▨	▨	▨
Implementation										
Enhancement and maintenance					▨	▨	▨	▨	▨	▨

▨ Indicates the extent to which each type of method covers each development phase.

for example, requirements analysis (Ethics) or project estimating (Slim).

- Some methods claim to cover the whole cycle (for example, LSDM/SSADM and Information Engineering) – but provide little assistance with project management or feasibility studies.

- Few proprietary methods incorporate techniques for package selection. In our sample, we found only four that did.

- Even fewer methods tackle personal (or end-user) computing. We found just two that gave some guidelines on how to manage such developments.

- Many proprietary methods can be used only to develop traditional mainstream data processing systems. This is because most methods are heavily data-oriented, whereas specialist systems (such as realtime or process-control systems) or, increasingly, systems built into products tend to focus on activities and time-criticality.

Thus, no one proprietary method is a complete solution to the demands placed on the systems department.

Methods need tools

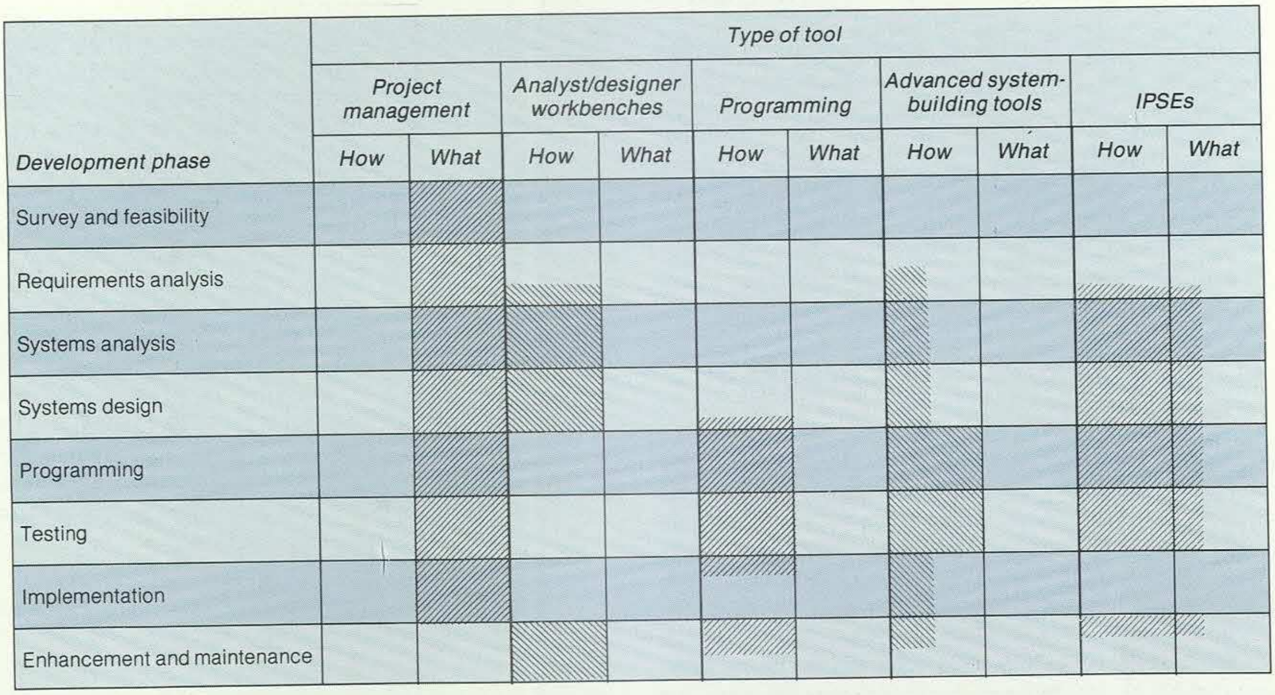
Some of the most valuable techniques employed by modern methods are either impossible or very cumbersome to use without suitable tools:

- Many project-management, planning, and estimating methods require easy access to and manipulation of large volumes of project data. Tools such as PMW (Project Manager Workbench), Maestro, and Prompt allow this to be done.

- Similarly, data-analysis diagrams will be prepared and maintained only if it is easy to do so. Tools such as an analyst/designer workbench are need to ensure this.

Our research suggests that where suitable tools are not provided, the method is regarded as tedious and time-consuming (even as impractical) and quickly falls into disuse. Conversely, if some types of tool are used outside the framework of a method, the underlying techniques are not understood, and the use of the tool flounders. The message here is simple. In implementing a method, do not forget to include appropriate tools. Figure 3 lists the range of facilities provided by system development tools. As with methods, no tool covers the whole of the conventional development process.

Figure 2 No one type of tool completely supports every phase of conventional development



▨ Indicates the extent to which each type of tool covers each development phase.

Introducing methods

Several decision have to be made in introducing system development methods:

Can you afford to wait?

Our research revealed that the methods and tools marketplace is still very immature. The methods and tools employed are at different stages of development; many are relatively new. We believe that over the next year to eighteen months there will be substantial improvements in tools and techniques and thus that any decision on methods and tools should be deferred if possible. Hence, the first decision involves analysing the current development problems and deciding if delay can be tolerated. The symptoms of acute difficulty include:

- Extreme user dissatisfaction.
- Wholesale lack of control of development costs and time.
- Significant quality problems.

Systems departments experiencing these symptoms cannot afford to wait.

Figure 3 Range of facilities provided by system development tools

Type of facility	*Tools providing appropriate facilities*
Administrative	Maestro provides facilities for electronic mail, standard forms, diary management, text handling, and version control of documentation.
Multi-user	Maestro, ISTAR, and BIS IPSE all provide a multi-user environment. (Note: Delta, Speedbuilder, and PDF naturally exist in a multi-user environment).
Project management	Artemis, Prompt, and PMW all provide extensive project management capabilities: – Artemis is most suitable for large projects. – PMW provides a high level of functionality but is PC-based (as is Prompt).
Requirements analysis	Core Analyst supports Core and is a sophisticated tool for requirements analysis.
System analysis/ design	Many tools provide system analysis/ design facilities. As yet, no one tool appears to be superior to the others, although Excelerator probably has the largest worldwide market share.
Implementation	Implementation tools vary widely, but IEF is the first tool to offer automatic code generation.

Which development process should methods be applied to?

As discussed earlier, it is important to understand the development requirements of the current and future (the next five years) application workload in order to determine the characteristics of the methods needed. Hence, the second decision involves determining which development processes need to be supported by methods – for example, conventional, iterative, small-systems, end-user, or application-package development.

Which development problems need to be tackled?

In the absence of methods or tools that cover the whole of the life cycle, organisations should only adopt such proprietary methods and tools that cover the most important parts of the development life cycle. One or two well-established, user-friendly techniques and associated tools directed at your most severe development difficulties will prove more beneficial than trying to configure a set of methods to meet all contingencies.

However, our research confirmed that most organisations find it extremely difficult to identify where the shortcomings are and what remedies are needed because their current development performance is not known. We believe that acquiring this knowledge is a necessary prelude to judging the comparative merits of proprietary development methods, and we would discourage organisations from choosing methods without it.

Which selection approach is the most appropriate?

There are essentially three possible approaches to selecting methods. The third decision involves deciding which is most appropriate in the light of the development processes.

Off-the-shelf: Some suppliers provide integrated methods and tools that they claim cover the whole development life cycle. Our analysis showed that this is not the case – but, there are nevertheless a few products (the integrated project support environments, or IPSEs) that come close to this objective. These integrated products are at an early stage of development and require extensive support. It is important to investigate carefully the commercial standing of some of the suppliers involved.

Build your own: This approach will provide the best fit – but constant updating will be required to take advantage of new concepts and software opportunities. It is very expensive and time-consuming and requires a level of expertise not commonly available in in-house development departments. (One organisation wrote off 25 man-years of effort before deciding to use proprietary methods.)

Configure your own: This approach involves mixing and matching techniques and tools and filling any gaps with internal standards so that the whole of the development life cycle is covered. Thus, the most appropriate products can be chosen – but there will invariably be some overlap between the selected products, which will need to be addressed.

Butler Cox recommends the 'configure-your-own' approach.

How do you choose proprietary products?

The choice of a proprietary method must be made by the senior management of the development department. The evaluation period can be kept short by concentrating on suppliers with a sound commercial basis and focusing on essentials such as the suppliers' commercial stability, product support (training, documentation, on-site assistance), and compatibility with the current situation. The latter is important since it includes not only equipment and software considerations but also cultural considerations. Large, bureaucratic organisations tend to prefer methods that emphasise control and documentation. Smaller companies prefer results-oriented techniques such as prototyping.

It is important to make the final decision as quickly as possible and to concentrate effort on introducing the method (or methods) properly.

We recommend selecting methods, with associated tools, from well-established suppliers. Short-list only those vendors whose commercial status and future are without doubt. Then spend significant time and effort in properly stage-managing the method's introduction and in policing its use so that the benefits are not frustrated by malpractice, and so that other methods are not introduced, by well-meaning staff, through the back door.

Critical success factors

For methods to succeed, time and effort must be spent on stage-managing their introduction. The critical factors for success include:

Senior business management consent

Senior business management agreement and support is necessary for several reasons. The costs, time, and lost opportunity attributable to introducing methods are so significant that corporate approval and backing is vital. Systems staff will be diverted from application work-in-progress, and their relationship with the users will change. Senior business management needs to understand and be prepared to accept these implications.

Systems department structure

New methods are likely to have a major impact on the roles of systems staff and may necessitate a departmental reorganisation. Do not begin the introduction of a method until this issue has been properly thought through and resolved.

Pilot the method

A pilot is the best way of deciding how best to implement the new method. Practical experience of the workings of the method in the environment and culture for which it is intended is enormously helpful in discovering the changes to existing practices and roles that may be required as a result of the method's use. Be careful about the composition of the pilot. Choose a project team that represents the average level of expertise and experience in your systems department. Your best staff are bound to make the method work – but you may learn nothing. Choose an application that is fairly typical – avoid high-risk projects, but do not select an application that is so trivial (or contrived) that it is unlikely to expose any difficulties. Above all, analyse the pilot results very carefully. You will need a thorough grasp of how the method is going to work in your environment.

Treat implementation as a major project

The implementation of a set of methods and tools to cover the whole development life cycle should be treated as a major project. Appoint a project manager and make him or her accountable for the success of the implementation – measurable in terms of effective use and gains in productivity. Ensure that the project team is properly funded and resourced (a typical team size is three or four staff) and that a formal implementation plan, based on the pilot results, is developed and agreed. This plan will contain, as a minimum, the training programme, the creation of guidelines and standards to complement the method, and the phased introduction of the method into both new development and maintenance projects.

Allow sufficient resources

Finally, it is worth stressing that the introduction of methods and their associated tools takes:

- Time (at least 12 months). Do not try to progress too quickly, and do not be surprised if the productivity improvements are not forthcoming immediately.
- Money. Do not skimp – especially on training.
- Good management. Staff motivation is vital. Use the pilot team as 'missionaries' for the new method.
- Support during the pilot, implementation, and even beyond. It is best provided by the supplier until implementation is complete. Establish an advisory service to provide continuing support for the method, to help development staff and users, to train new staff in the method, and to update the guidelines and standards complementing the method. Experience and changing circumstances will suggest the need for change. Figure 4 (overleaf) explains precisely what help you can expect from the suppliers (including what the deliverables are – that is, what you have actually bought) and at what price.

Monitor the payback

Our discussions with organisations that have invested in system development methods suggest that their motivation for so doing was an intuitive belief that methods help to improve the quality, cost, and timeliness of the development process. Whilst we would not quarrel with this belief, we would urge development directors and managers to adopt a more aggressive approach. The total investment in a method (especially once all the

Figure 4 What to expect from the method suppliers

Product feature	*Description*	*Pricing policy*	*Other comments*
Documentation	Manuals describing: – In general, the method and those phases of the development cycle it covers. – Step by step, for each major activity, what needs to be done and how to do it. – For each milestone, the deliverables (typically systems documentation), with worked examples.	This is the basic product, and it is normally sold at a fixed price, with, perhaps, some annual charge to cover updates.	Be aware that this documentation, which is normally voluminous, is on the suppliers' word processor. Any amendments will, therefore, be made physically by the supplier (irrespective of who prepared the amendment) and the client will incur word processing, printing, and distribution costs (often at alarmingly uncompetitive rates). It is possible to negotiate for the source documents to be converted to the client's word processor – with rights to amend it. This is also generally expensive.
Tailoring	Amending the method (especially the documentation) to fit the client's environment more closely. This might involve including any equipment or software restrictions, for example.	Time and materials at consultancy rates – apart perhaps for a limited number of days of free support included with the basic product. Word processing, printing, and distribution charges may also be applicable.	Tailoring generally must occur before implementation. As a consequence, the client is likely to have too little experience of the product to tailor it and will be dependent on the supplier. This is where the suppliers make their profit. Try to get a fixed price on this potentially uncontrollable expense.
Piloting	Assisting the project team to pilot the method through the first application.	Time and materials at consultancy rates.	Specific terms of reference are essential. Again, try for a fixed price. Be aware that the success of the pilot may be attributed to the supplier's facilitator – enthusiasm for the project may wane once he is off-site.
Training	Preparation and presentation of courses on the method for systems staff and users.	Fixed price per course – depending upon duration and location.	Another potentially large expense. Try to have courses on your own site; secure ownership of the material and become self-sufficient as soon as possible.
Ongoing support	Assisting systems staff after implementation.	Time and materials at consultancy rates.	Do not remain dependent on the supplier. Establish your own, in-house support unit.
Software	Providing software tools to support the method	Definitely optional. May be bought from a different supplier. In any case, pricing may be either once-off with an annual maintenance contract, or monthly rental. Any documentation (up to a specified limit) may be included in the software price – but not other requirements (like piloting, training etc).	Be aware that much of this software runs on microcomputers and the licence may be per machine. To make the software available to all users in the systems department can be very expensive.

support and training costs have been included) is high. You should look for and achieve a good commercial return on this investment. You should be prepared to set targets for improvements in:

- The rate of achievement of tested function points (or lines of code, for those who prefer).
- Error rates during all phases of testing and during initial implementation.
- Speed of implementation.
- Extent to which operational service agreements are not met because of development difficulties.
- The level of changes and enhancements.
- Maintenance costs (here, some organisations are suggesting that their maintenance costs have moved from 30 per cent to 5 per cent of the total development effort).

- Variations between actual expenditure and budgeted expenditure.

All of these improvements can and should be translated into real cost savings that you can use to justify the method in the first place.

Conclusions

The benefits that any specific organisation can expect will depend upon the scope for improvement, given its current development track record, the type of method introduced, and the attention and resources given to implementing the method. Hence, we re-emphasise our three main recommendations:

- Set up mechanisms to monitor your systems department's current development performance. Without this data you do not really know if you need to invest in new methods and you have no way of calculating (or measuring) the potential return.

- If you can, wait until the suppliers have produced better products. If you cannot wait, be sure to choose methods and tools that best meet your development profile and are supplied by mainstream vendors.

- If you invest in a method now, be sure to fund and resource its introduction properly – otherwise you will waste your investment.

Butler Cox is an independent management consultancy and research company specialising in the application of information technology within commerce, government, and industry. The company offers a wide range of services both to users and suppliers of this technology.

The Butler Cox Foundation is one of the services provided by Butler Cox. It provides the executives responsible for information systems in large organisations with a continuous analysis of major developments in the technology and its application.

The Foundation publishes six Research Reports each year together with a series of special Position Papers. The programme of activities includes a wide range of meetings that provide Foundation members with a regular opportunity to exchange experiences and views with their counterparts in other large organisations.

Butler Cox & Partners Limited
Butler Cox House, 12 Bloomsbury Square,
London WC1A 2LL, England
☎ (01) 831 0101, Telex 8813717 BUTCOX G
Fax (01) 831 6250

Benelux
Butler Cox BV
Burg Hogguerstraat 791
1064 EB Amsterdam
☎ (020) 139955, Fax (020) 131157

France
Butler Cox SARL
Tour Akzo, 164 Rue Ambroise Croizat,
93204 St Denis-Cedex 1, France
☎ (161) 48.20.61.64, Fax (161) 48.20.72.58

Germany (FR)
Butler Cox Deutschland Ltd.
Richard-Wagner-Str. 13
8000 München 2
☎ (089) 5 23 40 01, Fax (089) 5 23 35 15

United States of America
Butler Cox Inc.
150 East 58th Street, New York, NY 10155, USA
☎ (212) 486 1760 Fax (212) 319 6368

Australia
Mr J Cooper
Consultants (Computer and Financial) plc Australia
Level 5, 303 Pitt Street, Sydney 2000, Australia
☎ (02) 2870400, Fax (02) 2870450

Italy
SISDO
20123 Milano, Via Caradosso 7, Italy
☎ (02) 498 4651, Telex 350309, Fax (02) 481 8842

The Nordic Region
Statskonsult AB
Stortorget 9, S-21122 Malmo, Sweden
☎ (040) 1030 40, Telex 12754 SINTABS

Spain
Mr Sidney M Perera
Rosalía de Castro, 84-2°D, 20835 Madrid, Spain
☎ (91) 723 0995