Managing Contemporary System Development Methods

BUTLER COX
P.E.P

PEP Paper 6, June 1988

BUTLER COX
P.E.P

Managing Contemporary System Development Methods

PEP Paper 6, June 1988
by Norman Shipley

Norman Shipley

Norman Shipley is a Senior Consultant with Butler Cox who specialises in system development management. He has worked extensively with PEP on both the theoretical basis and the practical application of the metrics programme.

He has first-hand experience of system development, acquired from 1970 onwards in the course of working with major construction and mining groups, and in consultancy.

His early experience, with Costain, was in the development of simulation-based systems for production planning and company performance modelling.

With Taylor Woodrow, he was a manager with the central management services and systems department where he was responsible for various areas of technical and business system development and support. Over a period of 10 years, he managed developments ranging from systems for controlling mining and plant fleet operations, to systems for project financial control, estimating, and tender preparation.

With BICC he was IS manager for a subsidiary company where he managed the development and support of systems for overseas projects and for UK-based management and logistics functions.

His consultancy experience prior to joining Butler Cox included project-managing the retrieval of several serious accounting systems failures.

Norman Shipley is a chartered member of the Institution of Civil Engineers. He began his career in 1964, and for several years worked on various construction projects, where he became involved in the early use of computers for project planning and control.

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

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

BUTLER COX
P.E.P

Managing Contemporary System Development Methods

PEP Paper 6, June 1988
by Norman Shipley

Contents

BUTLER COX

Chapter 1

Managing contemporary system development methods

The management of system development projects has always been a problem. Project costs, commonly, have been more than expected, delivery dates have often been late, and the quality of the finished systems has sometimes not been good. But the adoption of structured development methods, allied to formal project management methods, supported by the appropriate tools, has helped to alleviate the problem.

The adoption of formal management methods has helped alleviate some of the problems of traditional development

However, such project management methods are mostly based on the traditional linear development cycle. System builders using the newer iterative methods of development (usually involving prototyping, fourth-generation languages, and so on), which we refer to as 'contemporary development methods', find that they are running into new management problems. In particular they have difficulty in scoping, estimating, and checking progress on the projects. These problems raise questions concerning the adequacy and appropriateness of the *management* methods and tools being used.

Additionally, some developers feel that a traditional formal management approach is in conflict with their perception of the advantages in speed and flexibility that contemporary methods can bring. They therefore face a dilemma in the choice of how far to follow, or abandon, a formal management approach. Their dilemma is more difficult to resolve in that there appear to be few ready-made methods that have been tailored to the needs of contemporary style development.

Developers may face a dilemma in how far to follow, or abandon, formal methods which do not fit

Users of contemporary development methods are convinced of the substantial benefits they bring. Their use will certainly spread. However, most users find these methods more difficult to manage. Furthermore, some of the hard-won lessons gained from managing projects using the traditional linear methods are in danger of being lost.

PURPOSE OF THE PAPER

We believe there is a balance to be struck between rigid planning and a completely free-wheeling approach to system development, and that the principles of project management should be the same, whatever the nature of the project.

There is a balance between rigid planning and a completely free-wheeling approach

The purpose of this paper is to explain *why* some of the management problems arise, and to show *how* management methods need to be modified to help remove them.

Development methods need to be supported by appropriate tools, or the expected benefits will not be realised. Management methods may also be usefully supported by management tools. We therefore also identify which characteristics and features of tools best match the needs of contemporary methods.

Chapter 1 Managing contemporary system development methods

We specifically do not attempt to prescribe the contemporary development methods to be used for undertaking system development — throughout we have concentrated on the management aspects of development. Nor do we attempt to justify the use of contemporary methods within this paper.

THE NEED TO IMPROVE THE MANAGEMENT OF CONTEMPORARY METHODS

The research for this paper, the PEP assessments completed to date, and our consultancy experience in the management of system development methods all confirmed there was a real need to improve the management of projects using contemporary methods.

The research specifically undertaken for this paper included:

- In-depth interviews with a number of organisations experienced in the use of contemporary development methods and advanced system building tools (specifically Mantis, Ideal, and their associated dictionary and database systems).
- Interviews with PEP sponsors regarding their experiences with contemporary methods.
- Interviews with suppliers of popular advanced system building tools such as Cincom (Mantis), ADR (Ideal), RCMS (Nomad).
- A brief survey of some of the available project management and estimating tools.
- A questionnaire survey of PEP sponsors regarding their use of project management methods and tools.
- An analysis of data within the PEP database to assess the effectiveness of using project management methods and tools.
- A search of relevant literature. The author also drew on his own personal experience in this area, and that of his consultancy colleagues.

CONTEMPORARY DEVELOPMENT METHODS GIVE SUBSTANTIAL BENEFITS

Development time and effort may be reduced by 50 per cent or more

In our interviews with organisations that have used contemporary methods (usually over a number of years) all reported substantial benefits in terms of reduced time and effort. This confirms the findings presented in reports from other consultancy studies and government- and industry-commissioned research. Besides the direct benefits on each project, our interviewees claimed other benefits related to managing the system development function as a whole.

Many of the published reports presented the savings in time and effort as deriving principally from the use of fourth-generation languages. But as we discuss later, there are other important system building tools that contribute to such improvements. Since we were not able to distinguish between the various sources in the data available, we treat all the benefits achieved as the result of a contemporary approach to development. In all of the results quoted below, percentage improvements are given in terms of total development effort, as compared with the use of traditional methods and tools. (In many cases Cobol or PL/1 were the traditional languages used.)

Reductions in timescale and effort
We asked users of contemporary methods for their own assessments of how timescale and effort compared with traditional methods, for projects of comparable size and complexity. The consensus was that time and effort reductions were of the order of 50 per cent, or more, and there were also reports of up to 80 per cent. Two organisations within the survey had actually developed the *same application* using both traditional and contemporary processes. Although in both cases the duplicate developments were done in the early stages of using the new methods (still within the learning period), the reported time and effort reductions were at least 50 per cent.

Earlier published reports in the literature also claimed reductions of the order of 50 per cent in timescale and 40 per cent in effort. Reductions in cost were less frequently quoted. Where they were mentioned, they were in line with the reductions in effort — as would be expected.

Benefits in managing the system development function
In addition to strictly *within* project benefits, a number of developers cited advantages which carry over into managing the *portfolio* of projects. These included smaller development teams, more flexibility in allocating staff to projects, and reduced maintenance effort and backlogs because the delivered systems are of a better quality. We refer to these benefits in more detail in Chapter 4 on page 27 onwards.

In addition there can be portfolio benefits

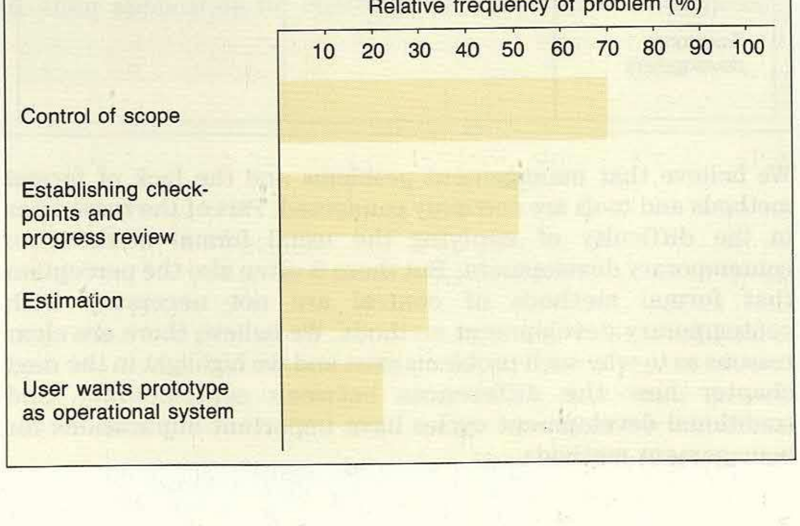
CONTEMPORARY METHODS ARE MORE DIFFICULT TO MANAGE
When asked to compare contemporary methods with traditional in terms of overall time and cost control, 80 per cent of the developers we interviewed said these aspects were *more* difficult to control.

We also asked them to identify the nature of the difficulty. As shown in Figure 1.1, the most commonly occurring difficulties were the control of scope, the setting of milestones and checkpoints for progress review, and estimation. Virtually all of the main management measures appear to be problematical.

Virtually all the main management measures are problematical

Figure 1.1 Most frequently occurring difficulties with contemporary methods

Virtually all of the main management measures appear problematic.



Chapter 1 Managing contemporary system development methods

SOME PAST LESSONS ARE IN DANGER OF BEING LOST

The need to use structured development techniques and professional project management techniques has been learned the hard way over a number of years. Since the early days of virtually undisciplined development, structured development methods, and to a lesser extent formal management methods, have become almost the norm in traditional development. Certainly this appears to be so amongst PEP sponsors, as our research shows. Recently, there has also been an explosion in the number of inexpensive planning packages, and the use of these is now very common.

But these lessons are in danger of being lost. We found that the use of structured development and professional project management methods is less common for projects employing contemporary development methods.

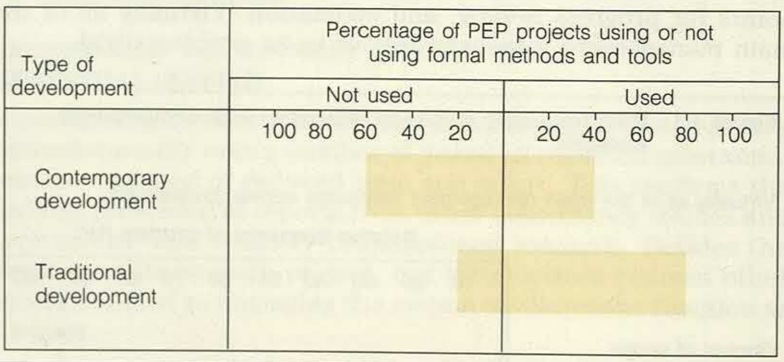
We asked PEP sponsors we surveyed to specify which formal development methods, management methods, and tools they used on their projects — and we then compared traditional-style developments against those conducted in the contemporary style.

The use of formal management methods is much less common with contemporary than with traditional methods

We found that the combined use of such aids, all of which contribute to formalising the development process, was noticeably less with contemporary development methods as shown in Figure 1.2. The combined use of formal methods and tools was only half as common in projects using contemporary methods. More generally only 40 per cent of developers use formal methods for contemporary development, whilst over 80 per cent use formal methods for traditional development methods.

Figure 1.2 Usage of methods and tools

Formal methods and tools are used less frequently with contemporary methods.



We believe that management problems and the lack of formal methods and tools are certainly connected. Part of the reason lies in the difficulty of applying the usual formal methods to contemporary development. But there is often also the perception that formal methods of control are not necessary with contemporary development methods. We believe there are clear reasons as to why such problems exist and we highlight in the next chapter how the differences between contemporary and traditional development cycles have important implications for management methods.

Chapter 1 Managing contemporary system development methods

Contemporary methods, as we have defined them, are still widely regarded as novel, but if the benefits claimed are the norm, then they are likely to become commonplace. At present we are in a period of transition, and a management approach is needed that will have the same consensus authority, and will afford the same stability, as that which has evolved for traditional development.

STRUCTURE OF THE PAPER

Project management methods, whatever the development style, must satisfy certain key business management needs. Business management needs:

- The impact on the business to be clearly related to any technical choices to be made and to be described in business terms (cost, timescale, resources...).
- A clear subdivision of a project into stages or elements, about which decisions are easier to make and that enable commitment of resources and cash to be made progressively. Risks can then be minimised.
- A clear statement of progress on each project and revised forecasts of future timescales, costs, and resource requirements at predefined review points. Decisions on whether or how to proceed can then be soundly based.

In practice, as we have shown, project management of contemporary methods seems not to be meeting these needs so well. Therefore, in order to improve the management practices, it is essential to understand what the differences between contemporary and traditional methods are. We analyse these differences in Chapter 2 and show how they lead to the various management problems.

Once the differences have been identified it is possible to suggest how the management difficulties may be overcome. This is the subject of Chapter 3.

Finally, in Chapter 4, we summarise the benefits that should be achieved from improving the project management methods as we suggest.

The Appendix identifies the features of some of the more common proprietary management methods and tools that are most relevant to their application for contemporary development methods.

Chapter 2

The new management problems in contemporary system development methods

There is now greater diversity in all kinds of system development methods and tools. There is also less standardisation than before, in terms of both methods, and the basic tools of system development, such as languages. So it is more difficult to analyse, improve, and modify development practice on the basis of common experience.

There is now a greater diversity in methods and tools — shared experience is more difficult to use

In spite of this diversity, a number of generally applicable conclusions can be drawn, and we discuss them in this chapter. They centre on changes that are occurring in three areas: the development cycle, the pace of development, and the resources engaged on development work.

The chapter begins by showing that management methods based on the traditional cycle are unsuited to contemporary development methods. The lack of a defined development method that has been modified to suit the changed development cycle is one of the principal causes of the current management problems.

It is not only the lack of a sufficiently clearly defined method that causes problems. There are also problems inherent in contemporary methods, especially with regard to prototyping. The chapter goes on to explain how the overlap between design and construction is making it harder to plan and control contemporary methods.

Next, the chapter discusses the changed pace of development. This has not affected all parts of the development cycle equally, however, putting a greater strain on management resources. We point out the changes that need to be made when planning and controlling projects.

Finally, the chapter discusses resources. There are important differences in the human resources involved in development. Also, whilst contemporary methods can deliver systems more quickly, using the wrong system building tools exacts a high penalty. Inappropriate use of analytical methods and techniques — and in particular the timing and use of data modelling — is a further problem area.

MANAGEMENT METHODS BASED ON THE TRADITIONAL CYCLE ARE UNSUITABLE

Whilst management methods do not change in principle, whatever the project, in practice they do not exist in a vacuum. In systems building the *development* techniques are closely linked with management techniques, and they provide the essential foundation upon which plans and controls are built.

In systems building the development method is the foundation for the management method

Chapter 2 The new management problems in contemporary system development methods

THE DEVELOPMENT METHOD IS THE FOUNDATION FOR THE MANAGEMENT METHOD

Project management methods are designed to satisfy the key management needs of planning and control (referred to in Chapter 1). In system development they have traditionally been achieved as follows:

- There is a plan that represents the project in (non-technical) business terms: that is in terms of objectives, deliverables, timescale, cost, resources, risk, and so on.
- The plan is not presented monolithically, but there are subdivisions of commitment and risk. Traditionally, this has been arranged by placing the major management review points at the end of feasibility, analysis, design, and construction. Typically, it is at these points that management has been able to exercise its prerogative of deciding to continue or abandon the project, or modify its scope.
- Between and at the phase-end points, the project plan is updated to depict changes of scope, current progress, and forecast cost and resource requirements to completion. This is normally achieved through time and cost recording, and by monitoring the completion of deliverables and the achievement of milestones.

The development method not only sets the framework for review points, but it specifies the basis for the project plan in terms of the activities and technical deliverables required, and in terms of activity sequence and dependency. Measurements of progress at any point, depend both on this method-defined framework and on there being a *baseline* of defined scope of work at every point.

The lack of well defined modified development methods is one of the principal reasons for control difficulties

Contemporary system development is based on a development cycle that is different from that used in traditional development. One of the principal reasons for the control difficulties experienced by many developers is the lack of defined *modified* development methods that adequately define the activities, deliverables, checkpoints, and milestones to be used in project management.

All of these factors conspire to make planning progress review and control difficult if based on the traditional cycle. In our research, we examined the difference between traditional and contemporary development cycles, to provide a basis for identifying measures to improve planning and control.

DEVELOPMENT CYCLE DIFFERENCES

During our research we examined the basic structural differences between traditional development styles and two prototyping approaches: 'throwaway' and 'evolutionary'. The results are shown in Figure 2.1 overleaf.

We looked at the differences in a number of areas as we go on to explain:

Difference in balance of work between phases

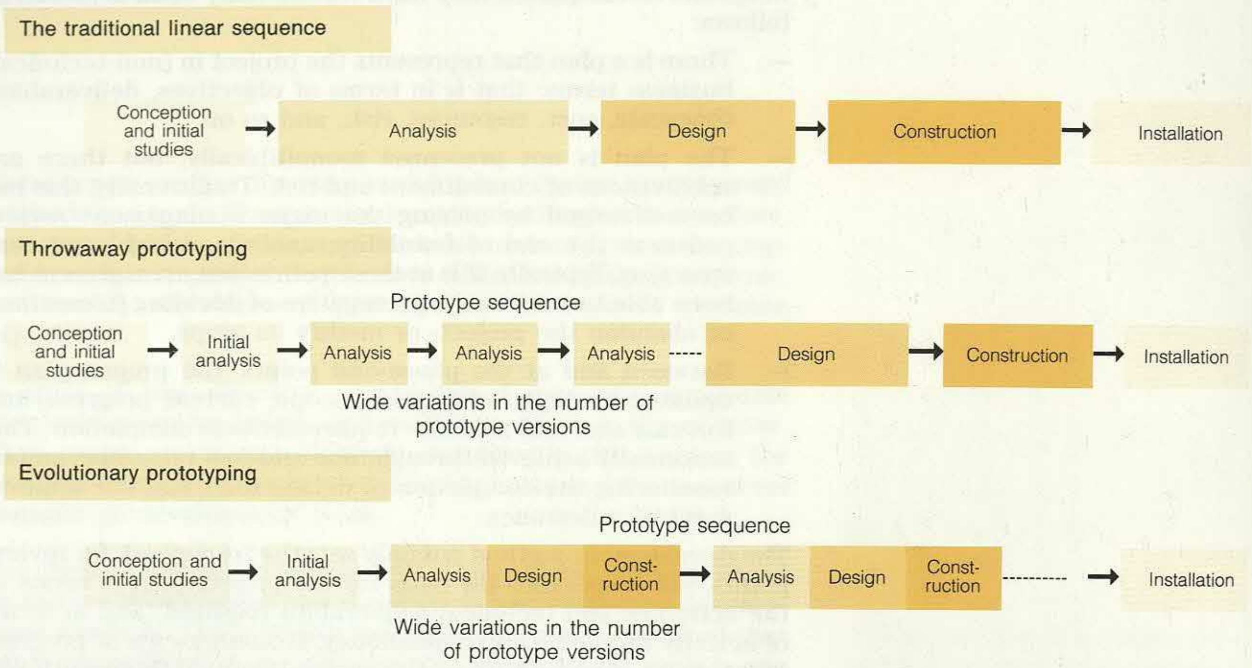
Although we could not compare traditional and contemporary methods over every part of the development cycle, we found large variations in the proportion of effort invested in what we call *initial analysis* in this report. This excludes any analysis done within prototype construction.

We found that the amount of effort invested in initial analysis varies between about 10 and 100 per cent of that which is typical for the analysis phase of traditional development. (We used

Chapter 2 The new management problems in contemporary system development methods

Figure 2.1 Development cycle differences

This figure depicts the differences in proportion, sequence, and stages between the various types of development methods.



information from the PEP database for comparison.) There are also large variations in the number of prototype cycles used. Figure 2.2 shows the variations we found from our project-by-project survey. Developers held different opinions over whether to pre-plan the number of prototypes to be used, or to carry out as many iterations as necessary. They also differed on whether to limit each prototype to certain design aspects (interface design for example), or to include all aspects of the design in every prototype. These results are shown in Figure 2.3.

Developers differ over how much preplanning and analysis should be done

Difference in sequence of phases

Even when classified crudely in terms of analysis, design, and system construction (regardless of actual distribution or repetition through the life cycle), the phases of contemporary-method development cycles showed significantly more overlap than in the traditional case.

We asked our survey respondents to estimate the overlap between activities in three areas: feasibility and analysis, analysis and physical design, and physical design and build. The results are shown in Figure 2.4 on page 10. Between analysis and physical design, and between physical design and build, the overlap is *two to three times greater* than with traditional methods. These overlap comparisons are indicators of relative timescale compression, and they suggest the extent to which control may, at the same time, become more difficult.

In practice, although nearly all developers said they planned a staged approach, there was almost universal acknowledgment that time pressures typically force developments into phase overlaps. With contemporary methods, there is more overlap and if it can be managed, such a highly concurrent approach dramatically shortens timescales.

If it can be managed, a concurrent approach dramatically shortens timescales

Figure 2.2 Number of prototype cycles

There is a wide range in the number of prototype cycles used.

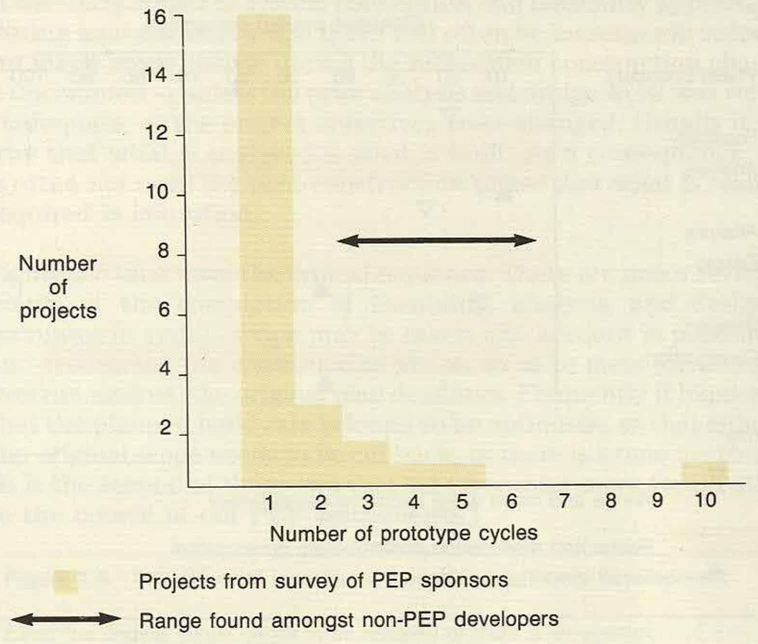
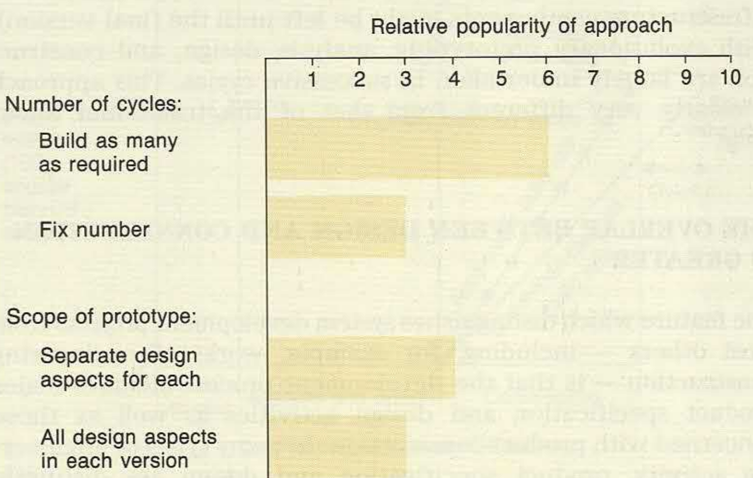


Figure 2.3 Different approaches to prototyping

Developers differ on how to approach prototyping.



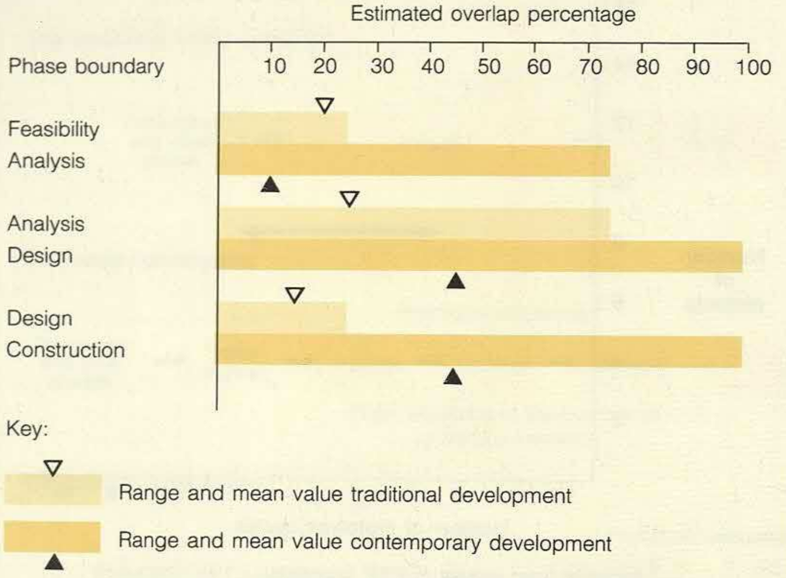
Different activities within each part of the cycle

Compared with the traditional linear model, the mix of activities in contemporary system development methods is much less homogeneous at any given point in the development cycle but the degree of difference depends, however, on the method of prototyping used.

With the *throwaway* approach, the predominant objective is to assist in requirements definition. Generally, the only products carried forward from one prototype version to the next are design concepts, and the main activity within the prototype building sequence is therefore analysis.

Figure 2.4 Phase overlap

With contemporary methods there is more phase overlap.



However, with the wholly *evolutionary* approach, the objective is to create a working system with each successive prototype version, such that physical deliverables are carried forward and refined until they are ready for installation (though certain infrastructure components might be left until the final version). With evolutionary prototyping, analysis, design, and construction are largely undertaken in successive cycles. This approach is clearly very different from that of the traditional linear cycle.

THE OVERLAP BETWEEN DESIGN AND CONSTRUCTION IS GREATER

One feature which distinguishes system development projects from most others — including, for example, works of engineering construction — is that the development process includes major product specification and design activities as well as those concerned with product construction. In many types of engineering activity, product specification and design are distinctly separate from construction and manufacture, and are usually carried out by separate teams having different skills. It is the separation of design and construction activities that eases project planning and control.

How design activities are controlled, and how well project schedules are matched (on a continuous basis) to the outputs of the design work, is a problem — particularly for prototype-based development. As our research showed, design and construction are much less separated with contemporary than with traditional methods. This has an important influence on the ease of controlling the development process.

How well project schedules are matched to the outputs of design work is a key factor in managing contemporary methods

Chapter 2 The new management problems in contemporary system development methods

DESIGN AND CONSTRUCTION WITH TRADITIONAL SYSTEM DEVELOPMENT METHODS

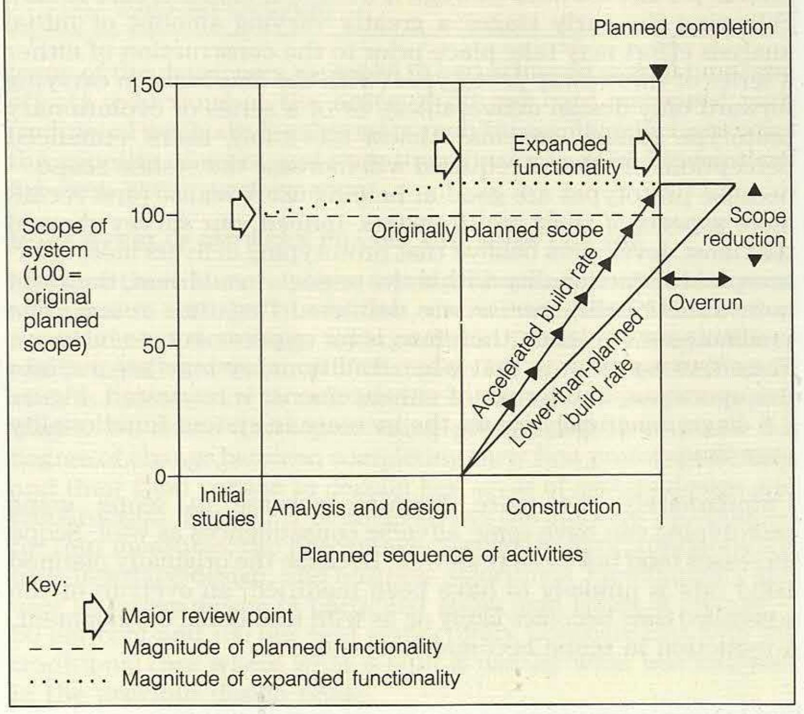
In traditional development — what is analysed is usually what is built

Traditionally, system scope is (theoretically at least) determined in the early stages of system conception and feasibility appraisal. During analysis and design there will often be increases in scope, but major scope change during the subsequent construction phase is uncommon — unless the prior analysis and design work was very inadequate, or the project objectives were changed. Usually it is true that what is analysed is what is built. As a consequence, it is often not until the post-construction phase that what is really required is identified.

Figure 2.5 illustrates the typical sequence. There are major review points at the completion of feasibility, analysis, and design. Increases in system scope may be taken into account in planning and resourcing the construction phase, so as to meet (or accept overrun against) the original plan deadlines. Frequently it happens that the planned build rate is found to be optimistic, so that either the original scope needs to be cut back, or there is a time overrun. (It is the second of these two that we encounter more frequently in the course of our PEP assessments.)

Figure 2.5 The effect of increase in scope in traditional development

Either the original scope needs to be reduced or there is an overrun.



When using traditional development methods, overrun is usually less to do with a change in scope than it is with underestimated construction effort (although changed requirements play a part, of course). This is often because important parts of the design (in particular data structure design) are carried out by the coding team during construction, particularly when third-generation system building tools are used in the absence of a rigorous prior physical design plan. The effects of this are consistent with Putnam's theoretical models used in PEP. (See PEP Paper 5 for a description of them.) The basis for these models is the premise that both within-team communication, and coordination of design and other information, increase dramatically in difficulty with system size (including increased scope) and time pressure. Even when there are no significant increases in scope, these difficulties tend to be underestimated.

There is nonetheless an opportunity at the completion of analysis and design to incorporate increases in scope, or other implications of the design outputs, into subsequent plans.

DESIGN AND CONSTRUCTION WITH CONTEMPORARY SYSTEM DEVELOPMENT METHODS

The management of design is a particular problem in contemporary method management. For example, one of the key benefits claimed for prototyping is that it enables users to communicate their system requirements more effectively than traditional paper-based specifications, whether they are described in verbal or analytical-model terms. So prototyping facilitates earlier and better perception of what users really require. But that virtue can also have undesirable consequences too if the design process is not well managed.

Prototyping enhances perception of requirements — but there are disadvantages

We found that scope definition early in the development cycle may often be inadequate when using prototyping, making subsequent effort harder to direct and control. This serves to exacerbate the effects we discuss here (though it does not, alone, create them). Following the early stages a greatly varying amount of initial analysis effort may take place prior to the construction of either a series of throwaway prototypes (with the emphasis on carrying forward only design deliverables), or of a series of evolutionary prototype system versions. Almost inevitably, users' enhanced perceptions of what is required will increase the system scope — because prototypes are good at helping users realise (and recall) more aspects of their requirements. Indeed, our survey showed that most developers believe that prototyping delivers more user-acceptable functionality within the project. In addition, they said more functionality *per se* was delivered first time round. The predominant tendency, therefore, is for requirements to increase. The effect is similar to that when traditional systems are put into live operation, and a rush of enhancements is requested. Figure 2.6 diagrammatically shows the increase in system functionality over time.

The tendency is for scope to increase but corrective (planning) action may not be taken

Unfortunately the more gradual increase in scope using prototyping can have some adverse consequences as well. Scope increases tend not to be reviewed. Because the originally planned build rate is unlikely to have been modified, an overrun of the scheduled time becomes likely or, as with traditional development, a reduction in scope becomes necessary.

THE PACE OF DEVELOPMENT IS GREATER

As well as overlapping phases, the heightened pace of development is leading to additional management problems.

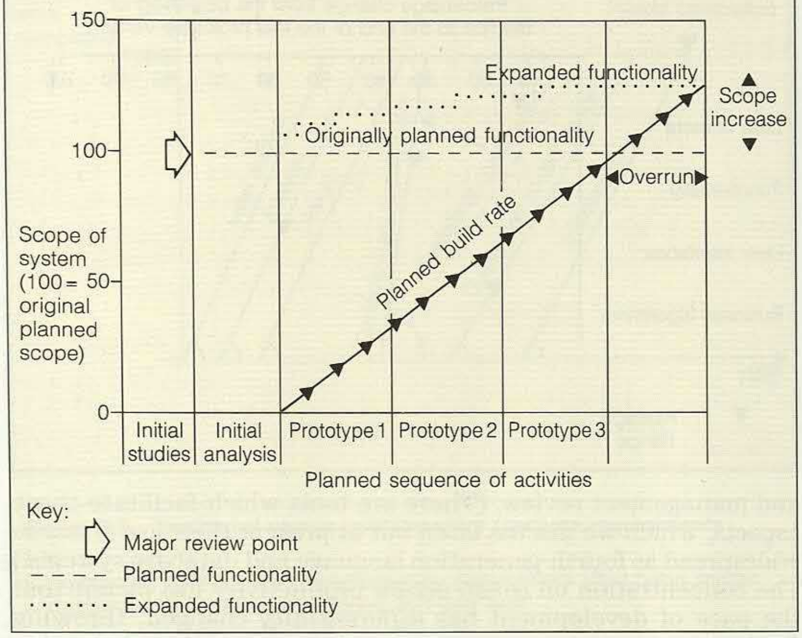
REDUCTION IN DEVELOPMENT TIMESCALES

A number of factors combine to throw more demands on project managers and the management methods and tools they use. The most obvious factor is timescale reduction. As mentioned earlier,

The increased pace of developments is leading to additional management problems

Figure 2.6 The effect of increase in scope in contemporary development

Prototyping facilitates communication of requirements and leads to an increase in scope. However, the necessary corrective actions to maintain original delivery dates or control additional functionality are often not taken.



many of the developers we spoke to had achieved overall time (and effort) reductions of the order of 50 per cent compared with traditional methods. Furthermore such timescale reductions whet the appetite of users, and raise their future level of expectations for even better performance.

MORE OVERLAP BETWEEN PHASES AND MORE CHANGE

Because there is more phase overlap, there are more concurrent activities to manage. In addition, there is evidence to show that there is more to manage and control in terms of *change*. There can be significant amounts of change in key areas of system design, such as functionality or user interface, throughout the prototyping stages. We asked developers using prototyping to estimate the degree of change between completing their first prototype version and their final version in certain key areas of system design and construction. Figure 2.7 overleaf shows the answers we obtained for four measures: the percentage changes in data, functionality, user-interface-design, and business algorithms. Functionality and user interface design in particular show high levels of change — 50 per cent and 100 per cent respectively. Compare this with the traditional case where what is built is usually what was analysed in the previous design phase.

Such changes make it difficult to control adjustments to timescales and resources, or even to record progress.

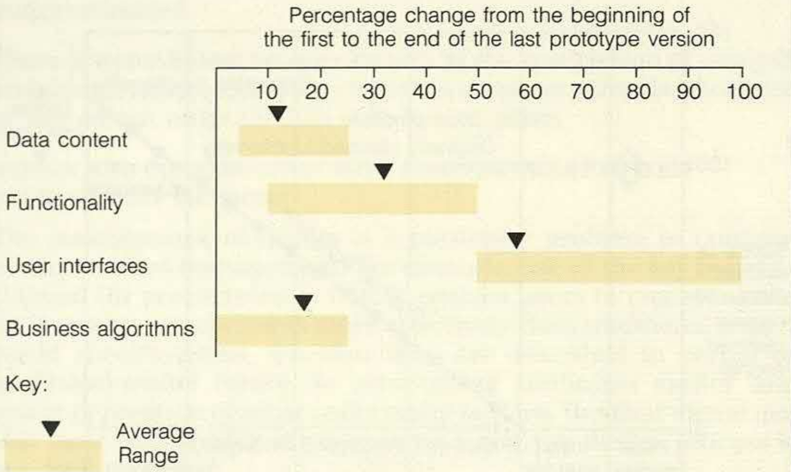
THE BALANCE OF RESOURCES IS DIFFERENT

The pace of development has differentially changed, throwing further strain on management resources

System building tools have given benefits predominantly in the construction component of development. There has been less benefit in terms of speeding up the process of design, and quality

Figure 2.7 Degree of change in system design using prototyping

Specification of required functionality and the definition of the user interface tend to change by 50 to 100 per cent between the first and last prototype when using successive prototyping.



and management review. (There are tools which facilitate these aspects, which we discuss later, but at present their use is not so widespread as fourth-generation language and database systems.) The concentration on *construction* productivity has meant that the pace of development has *differentially* changed, throwing further strain on management resources, as we show below.

There are also a number of other resource-related aspects that differ from those in traditional development, both because of factors relating to pace and because of some of the inherent properties of prototyping. All of these differences impinge on planning and control, and we discuss each of them below.

DIFFERENT BALANCE BETWEEN DEVELOPMENT PHASES

In planning terms, the original *line of balance* between the main development activities is disturbed when moving from traditional to contemporary methods. Compared with the traditional case, both design and review activities now lag behind construction, and become new bottlenecks. These effects are illustrated in Figure 2.8 which depicts the development cycle in terms of a simple six-activity model.

Both design and review now need relatively more management involvement than the construction activity. This points the way to the most profitable future areas for applying advanced system building tools, and to the new balances of project resources that need to be found.

Design and review activities now lag behind construction and can be the new bottlenecks

SMALLER TEAMS OF A DIFFERENT COMPOSITION

In our survey, we asked developers to specify the types of skill they employed across all parts of the development cycle. There was a strong contrast between traditional and contemporary methods, as shown in Figure 2.9. Contemporary methods use less resources. Also, the resources needed tend to be concentrated more into a single-system developer role rather than being spread across all the system development skills. As a consequence, resources are generally more interchangeable between projects. This has implications for portfolio management, as we discuss later.

There has been a convergence of skill types towards the single 'system developer' role

Chapter 2 The new management problems in contemporary system development methods

Figure 2.8 The balance of work and resources

To complete a project in minimum time each part of the project cycle should proceed at the same rate. With modern (contemporary) methods, the balance of work and resources has changed.

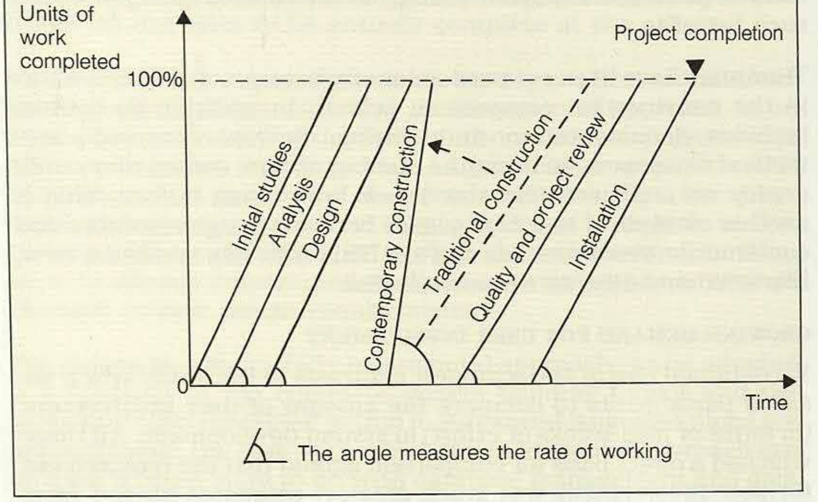
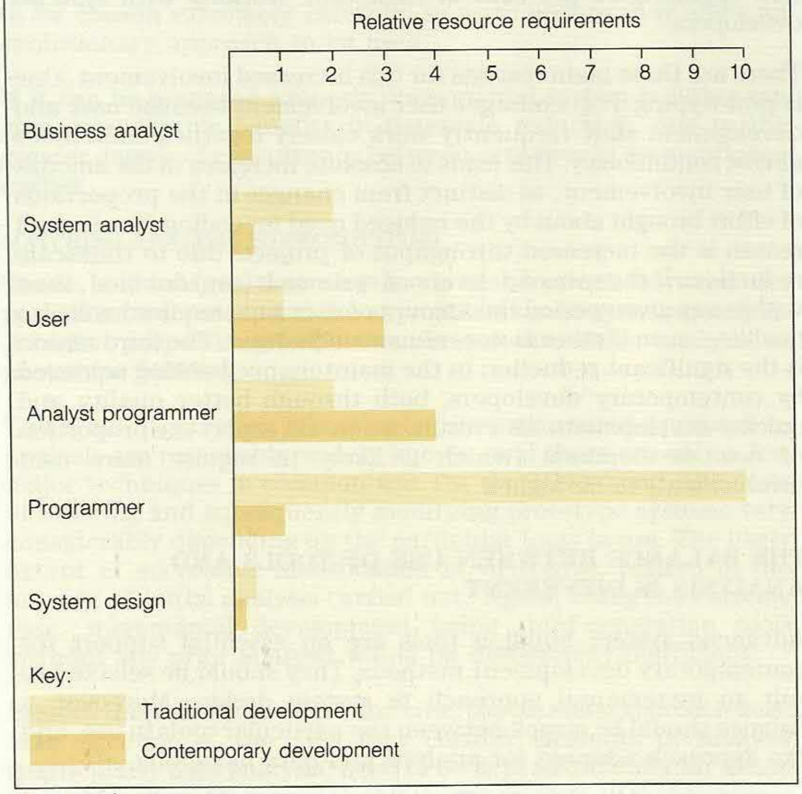


Figure 2.9 Traditional versus contemporary resource requirements

Contemporary methods are characterised by development effort being smaller and provided by 'multirole' developers and user staff.



Contemporary developers were unanimous in claiming that their project teams were smaller — only 50 per cent, or even less, of the size needed in traditional development.

Chapter 2 The new management problems in contemporary system development methods

The quantitative models used in PEP demonstrate the inefficiencies of larger teams and of intense manpower buildups. Therefore the consequence of these smaller team sizes is that there should be more productive working. This gain should be independent of any gains deriving from the use of system building tools — provided that poor management control does not erode such benefits.

Team sizes have been cut predominantly because of reduced effort in the construction component, which, in addition to coding, includes documentation and testing. Several contemporary-method developers said that the number of pure coders they could deploy on a project was now much lower than before. This is another example of how the balance between design resources and construction resources has shifted. This shift has created a new, scarce commodity, as we now discuss.

Development teams are much smaller

GROWING DEMAND FOR USER INVOLVEMENT

We obtained one of the strongest contrasts in the study when we asked participants to compare the amount of user involvement (in terms of man-weeks of effort) in system development. All those who had a direct basis for comparison agreed that the requirement for user involvement was much greater: estimates ranged from *two* to *four times* as much as in traditional development.

One developer told us "if the user is not involved much more, then the methods are not being used properly". Another added that, in his organisation, "some senior user representatives are now spending 80 per cent of their time working with systems developers".

In some organisations the user is now the scarce resource

There are three main reasons for this increased involvement. One is prototyping. It encourages user involvement because user and development staff frequently work closely together, sometimes almost continuously. This leads to absolute increases in the amount of user involvement, as distinct from changes in the proportions of effort brought about by the reduced need for coding. The second reason is the increased throughput of projects due to timescale reduction. If the rate of delivery of systems is (say) doubled, then within any given period the amount of user time required will also double — even if other factors remain unchanged. The third reason is the significant reduction in the maintenance backlog achieved by contemporary developers, both through better quality and quicker development. As a result, we would expect the proportion of *new* development (which is likely to require more user involvement) to be higher.

THE BALANCE BETWEEN USE OF TOOLS AND ANALYSIS IS DIFFERENT

Advanced system building tools are an essential support for contemporary development methods. They should be selected to suit an incremental approach to system design. Moreover, a balance should be struck between the particular tools in use, and the approach adopted for analysis and data modelling.

MATCHING TOOLS TO THE INCREMENTAL APPROACH TO SYSTEM BUILDING

Suppliers claim that advanced system building tools, and particularly tools featuring design dictionaries and relational database systems, have properties of application independence, ease of system modification, and maintenance. These properties, they say, allow a completely incremental (that is to say evolutionary) approach to system building, in which data and function may be added progressively. The incremental approach, it is claimed, avoids excessive penalties in terms of rework, because analysis and design do not have to be entirely complete at the outset.

Choose system building tools which allow an incremental approach to system building

Almost all of the respondents in our survey confirmed that their system building tools *did* allow an incremental approach to be used. The estimated reduction in the rework incurred with advanced system building tools, against that required with traditional third-generation tools (such as Cobol, 'flat' file structures, and so on) ranged from 50 per cent less to as little as five per cent. (The comparison is for equivalent amounts of modification to already constructed modules or to systems as a result of changed or new design requirements.)

If some elements of the toolkit are missing extra unplanned work may be introduced

The degree to which a fully incremental approach can be adopted is a function of the complete system building toolkit and the particular tools contained within it. Advanced languages alone are not sufficient; the ease with which the incremental approach can be used is often more to do with database management and data dictionary systems. If these, or other, important elements of the toolkit are missing, extra *unplanned* work will be introduced, disrupting the project schedule. In the extreme case, if third-generation tools were to be used with a wholly evolutionary approach, the modifications and work involved would get out of hand after only a few iterations. System building tools thus need to be chosen extremely carefully so as to ensure they allow an evolutionary approach to be used.

If it can be managed properly, incremental system building can give considerable benefits in timescale reduction, due to the greater degree of concurrency between activities as we mentioned earlier.

MATCHING ANALYSIS TO PROTOTYPING

The relative costs of analysis and prototype building depend on the tools used

There is a balance that also needs to be struck between the particular tools used, and the approach taken to systems analysis. A higher proportion of effort can be put into either analysis, or prototype building.

The costs of adopting a comprehensive structured approach to analysis are reasonably widely known, and there are only a few major techniques in common use. On the other hand, the costs of building and subsequently modifying prototype systems vary considerably depending on the particular tools in use. The likely extent of successive modification is, of course, related to the amount of initial analysis carried out. Again, citing the extreme case, incremental development using third-generation tools without any prior analysis would be disastrously inefficient.

Data analysis must lead

Besides this balance of effort and cost, the planning approach must take account of the fact that certain elements of analysis (particularly data analysis) need to be kept sufficiently far ahead of construction. We return to this theme on page 19 in the context of enterprise modelling and project portfolios.

Chapter 3

Improving the management of contemporary system development methods

Development methods and tools, and project management methods and tools fulfill different purposes. The former are mainly concerned with *how* the development process is to be executed; the latter are mainly concerned with *what* needs to be done and *when*. But, as we have explained, they need to be very closely coordinated because of the strong link between the management method and the development method used. To improve management control, project management methods need to be adapted to suit the contemporary methods in use. Both planning and system building tools also have a part to play.

We believe that there are five aspects of system development that need to be addressed:

— The development methods used should be more formally defined.

— The estimating and planning practices should be modified to suit the changes in the development process.

— The procedures used for reviewing and controlling progress should then be based on the milestones and targets set by the more formal development method.

— Appropriate system building tools should be used to support the increased rate of development made possible.

— Portfolio management should be improved by taking advantage of within-project benefits such as smaller development teams, and more interchangeable resources.

Furthermore, the experience gained in the use of contemporary methods should be consolidated so that future planning and control may be more rapidly improved.

FORMALISE THE DEVELOPMENT METHOD

Formalise the development method to give a sound basis for project management

As we stated previously, the development method forms the basis for the project management method. If it is not clearly defined and adhered to then there is no clear basis for project management tasks such as planning, estimating, or control. So, the first need is to define more formally the contemporary method(s) in use. This will promote a common understanding both within the system development function and also with the system users, thereby forming the basis for a sound project management method.

The development method needs to allow for defining the system scope clearly and in advance of prototyping; defining clear milestones and checkpoints, taking account of the trade-off between analysis and prototyping; taking advantage of the features offered by system building tools; concurrency and overlap of development activities; and variations between different projects.

Chapter 3 Improving the management of contemporary system development methods

DEFINE THE SCOPE CLEARLY AND IN ADVANCE

Adequate scope definition in the early stages is important

Adequate scope definition in the early stages is particularly important. Whilst prototyping may help to identify which system features are most valuable, there are distinct dangers in relegating too many decisions to the users most intimately involved in building and reviewing prototypes.

There can be great merit in traditional-style written specifications, provided they are succinct and at the right level. Examining functionality through the medium of a series of screen formats can be mesmerising, and can sometimes obscure fundamental design and value-for-money matters. Traditional-style specifications can ensure that matters are raised and decided at the right level, and they can be used to supplement the prototyping process.

In traditional system design, the feasibility stage is in reality the first major planning phase. With contemporary methods it is even more important to build an adequate scope definition into the early stages of the development sequence — otherwise an early management opportunity will be lost.

DEFINE CLEAR MILESTONES AND CHECKPOINTS

A lack of clear milestones and checkpoints was frequently cited as a problem in our survey. This shortfall stems, in part, from an insufficiently defined development method (it is also due to milestones being wrongly defined and blurred by the overlap of phases to which we have already referred).

Milestones and checkpoints need to be clearly defined in terms of revised deliverables

There should be an explicitly defined development sequence that both positions any iterative sequences within the overall framework and that extends to defining major activities within the prototyping sequence itself.

The modified development method should contain a clear definition of the revised deliverables that are required. Typically this will involve both the deletion of deliverables normally found in traditional development projects, and the substitution of others.

TAKE ACCOUNT OF THE TRADE-OFFS BETWEEN ANALYSIS AND PROTOTYPING

As we have already noted, there is a balance to be struck between analysis and prototype construction. The most cost-effective balance depends on the particular system building tools used, and this should be reflected in the development method sequence.

Data analysis and functional analysis need to be planned differently

In particular, there is a difference between data modelling and functional analysis. Whilst system building tools may allow data and function to be added piecemeal, in human terms it is difficult to add and integrate data relationships in this way. Function can be handled more discretely, and it is easier to analyse and add piecemeal.

Modified development methods should therefore take explicit account of the balances involved, and should ensure that data modelling remains well in advance of design and construction. This balance also applies, though on a larger scale, to a series of development projects or the whole application portfolio. Enterprise data modelling should also occur well ahead of preparing the application development plan.

TAKE ADVANTAGE OF SYSTEM BUILDING TOOLS

Development methods should be modified to take account of some of the features of the system building tools used.

Modify the development method to take advantage of the system building tools used

To begin with, it helps to delete unnecessary deliverables from the development method. For example, with certain system building tools, many developers find it unnecessary to produce traditional style program specifications, because of the self-documenting properties of the tools.

Many more powerful features than this one are available, but it exemplifies how the development method and standards should be modified to take full advantage of the productivity gains on offer.

ALLOW FOR CONCURRENCY

The development method should allow for concurrency

Because contemporary system development is, by definition, non-linear, the method should allow for concurrency. Moreover, it should take advantage of it — after allowing for the difference between data and functional analysis. The development method should, for example, provide guidance on post-construction integration of subsystems built in parallel.

ALLOW FOR VARIATION BETWEEN PROJECTS

Tailor the exact sequence to the needs of the project

A key characteristic of contemporary development is its flexibility. In a sense, this flexibility contrasts with the requirement of a defined development sequence for planning and control. The defined sequence includes the nature and number of prototypes to be built. So the exact sequence of events should be tailored to the needs of each project. For this reason, and also because of the changed pace of development, it is preferable for the development method to be held as a *modifiable template* in the form of a software package.

MODIFY ESTIMATING AND PLANNING PRACTICES

You should choose estimating methods and tools that can cope with changes in phase sequence and resources. It is a mistake to use unmodified traditionally-based measures and estimates.

Do not use traditionally-based estimates

USE APPROPRIATE ESTIMATING METHODS

There are two broad classes of estimating method. The first is macro estimating, which is based on high-level descriptors of the characteristics of the system to be developed (such as application type, number of subsystems, number of user departments involved, and logical inputs and outputs). The second is task-based estimating, which is useful when individual project tasks or activities can be identified, and when at least their approximate size and content is known.

Both have a place. There is no alternative to macro estimating in the early stages of a project. Task-based estimating comes into its own when planning takes place at several levels, or when planning is very short term (for example, over a period of just a week or two ahead). The two approaches should be used in combination, so that estimates made at the task level reflect the macro estimate in terms of overall resource requirements and duration (unless there are good reasons to modify the earlier forecasts).

Chapter 3 Improving the management of contemporary system development methods

In our survey we asked developers to specify where in the development cycle they normally made their estimates, and what they believed their estimating accuracy to be at each point. We found little difference in the accuracy of estimating at the feasibility stage: most estimates were uniformly inaccurate, for both traditional and contemporary methods. However, most developers claimed that the accuracy of estimates for subsequent technical design and construction activity made at the end of functional design were much improved. Where there was a direct basis for comparison, developers suggested that estimating errors were less — a maximum of about 10 per cent as compared with 25 per cent for traditional construction methods.

Contemporary methods have the potential to be better controlled than traditional methods in terms of time and cost

These results indicate that, with contemporary methods, the risk of estimating inaccuracy is generally lower *earlier* in the cycle than with traditional methods. This suggests that contemporary methods have the potential to be better (not worse) controlled than traditional development, in terms of cost and time overrun. The fact that this statement is at variance with developers' experience points again to the likelihood of problems in management and control.

USE ESTIMATING TOOLS

Software-based estimating tools are more complex and far less common than planning and control packages. The fundamentals of their operation are usually less visible, and less well understood, than with other types of tool. They should be suited to the particular development methods you intend to use. If they also have automated links with, for example, development method templates and planning packages, they can lead to savings in management time derived from making plan generation more automatic.

Estimating tools should suit the development method in terms of phase sequence and proportion, productivity factors and resources used

There are other features to look out for. One is the model on which an estimating tool is based. Whether a formal mathematical model or a simple rule-based type it should allow modification for contemporary method development in terms of differences in phase sequence, reductions in timescale and effort, and changes in the types of resource required for the project. Estimating tools that work on the basis of mathematical models derived from statistical analysis of completed project data can give valuable predictive results. In general they will be more soundly based than heuristic methods derived solely from the relatively small sets of data that an individual user has available for analysis. But, in any transition from traditional to contemporary methods, the ability to calibrate estimating models to the productivity characteristics of the developer's own environment is essential.

Another feature to look out for is the basis of the system development method. Many task-based estimating models are based on a particular (proprietary) system development method. You should check that the method is suitable for your needs.

Estimates should be progressively improved as development proceeds

A further important feature is the ability to progressively improve the estimate as the project proceeds. Some tools allow actual performance data to be accumulated and used to refine project estimates. Furthermore, such data is valuable for future analysis and refinements to the estimating methods.

In the Appendix we identify some popular estimating tools and summarise their principal features.

Chapter 3 Improving the management of contemporary system development methods

PREPLAN THE SCOPE AND PROTOTYPE DELIVERABLES

You should be able to plan more tightly by better defining the development cycle, by adequately defining the system scope at the outset, by defining adequate milestones and checkpoints, and by adopting some simple measures.

One such measure is to specify the number of prototype versions to be built. Each can be assigned an individual scope, which is a prescribed part of the total functionality required. Whilst these divisions may not remain constant throughout the project, they will form a solid baseline. Another measure is to set preplanned milestone dates for the completion of each prototype version, based on estimates of work content and resource requirements.

Preplan the number of prototypes and predetermine the scope of each

The main reason for defining these measures is to enable better progress review and control, a topic that we discuss in more detail beginning on the next page.

CHOOSE PLANNING TOOLS THAT MATCH THE CHANGES IN DEVELOPMENT PACE AND RESOURCES

One of the reasons why planning methods and tools are less commonly used with contemporary than traditional methods is because of the former's faster pace of development. Developers genuinely find difficulty in preparing and updating plans quickly enough. But the planning process can be speeded up in a number of ways through the use of planning tools.

The changed pace of development means that planning needs to be speeded up

An example is a modifiable system development template — one that can be tailored to the needs of a particular project. Some of these templates enable the plan basis so created to be transferred directly into a planning package. Since plan formulation is one of the most time-consuming parts of planning, considerable savings can be made in this way. Alternatively, even plan templates, stored and modified only within the planning package itself, will save time, and will act as checklists.

For these approaches to work, the planning package needs to be able to handle dependency logic between activities — so that the plan may be scaled up or down according to the actual durations without distorting the sequence.

Using the development method template in conjunction with an estimating tool represents a further refinement. Activity content, sequence, and duration may all be manipulated together to form a complete first-cut project plan, which again can be transferred to a planning and control package.

Because most system developers and users are not professional planners, planning packages should be easy to learn, and have good quick-start manuals or software help guides. Increased user involvement means that more people need access to project plans, so they should be shareable and planning packages should be chosen particularly for their capabilities as a medium for communication. Bar chart representations, for instance, should be visually clear and easy to follow. The variety of symbols demands good screen design, and some representations appear arcane and cluttered. The same comments apply, though even more strongly, to project network representations. In both cases, it should be possible to scroll through the complete plan on a display screen.

Planning packages should be good communication tools

It is a distinct drawback when plans have to be printed before the effect of a change or update can be seen.

USE PLANNING TECHNIQUES APPROPRIATE TO THE DEVELOPMENT METHODS

For the software building component of a development simple control techniques can be very effective

Planning techniques also need to be selected in relation to the development method. Network logic and critical-path facilities are both required, but often they are more useful for analysing the complete development (which may include equipment selection, procurement, training, and so on), than merely the software development component. For that, simple techniques are usually more useful than, for instance, status reporting in terms of criticality, float, and early and late starts and finishes. Within each sequence of prototype construction, the main deliverables are easy to identify given an adequate initial scope list. Typically they will be the completion of particular business functions, individual programs, screens, reports, tested modules, and so on. At this level of the plan, within a given subdivision of the whole system, there is often no unique work sequence attached to completing a set of (say) functions or programs. Often they can be tackled in any order. One of the simplest devices for tracking progress is simple *rate-charting* of the completion of each set of deliverables against time. This may be used to give an immediate indication of the production trend.

At the project level *standard-spend* curves can be used, which are based on the same simple philosophy. Given the speed of prototyping this type of technique can often be more practical than complex network updates.

Wherever possible, progress to date in the form of milestone achievement, and extrapolated production trends should be displayed graphically. It helps if the data-entry procedures for flagging completion of activities, or sets of activities, are kept as simple and as quick as possible. Re-estimates need to be made at regular intervals to take advantage of the fact that estimating accuracy improves more quickly with contemporary methods. The planning package should report all incomplete tasks for re-estimation, and then permit global changes of resource levels, or planned production rates, to be incorporated.

MODIFY PROCEDURES FOR REVIEW AND CONTROL

The milestones and checkpoints obtained by formalising the development method, and by preplanning the scope and deliverables, may now be used for reviewing and controlling progress. The deliverables and delivery dates fixed in the plan form the baselines against which to measure progress. (In the absence of such simple measures, it is hard to exercise control, because for any given date there is no prescribed deliverable, and for any given deliverable there is no corresponding date.)

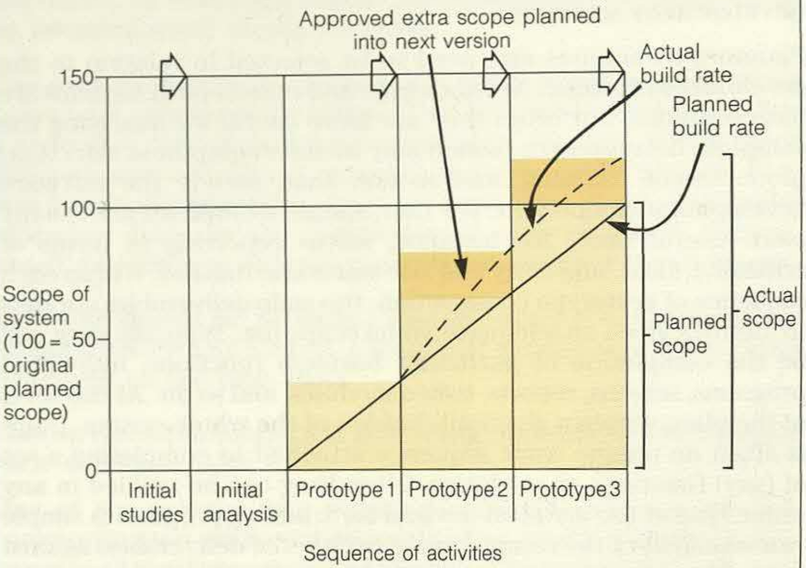
Successive prototype versions are the analogy to the traditional phase-end points — when used properly for project review they can provide clearer decision points

Successive prototype versions provide the analogy to the traditional phase-end review points. If used properly, they should provide clearer decision points — but only if, at the point of review, there is a preplanned set of deliverables derived from an earlier allocation of system scope. Figure 3.1 depicts how planning and control measures can work together.

Chapter 3 Improving the management of contemporary system development methods

Figure 3.1 Project control during prototyping

Prototyping allows the scope of the system to be increased in a controlled way.



INSTITUTE PROPER CHANGE CONTROL PROCEDURES

One useful approach to change-of-scope requests takes the following form:

— First, change requests are checked against the scope of subsequent planned prototypes to ensure that they are not already included.

— Next, change requests that give rise to unplanned additions to the scope are passed through an appropriate approval procedure.

— Finally, approved changes are *not* attempted in the current prototype (unless it is the final one) but instead are planned for the next or a subsequent prototype.

This approach helps encourage planned additional work, through the greater separation of design and construction, and by forcing more explicit review of design outputs. Placing extra work in a subsequent prototype version provides the opportunity to modify the resources and schedule if necessary — much as in the traditional case, where there is naturally more separation between design and construction.

Change control procedures should help separate design from construction and encourage planned additional work

THE TIMING OF REVIEWS

Just how the main prototype-end review points are used and arranged depends very much on the scale and complexity of the project. They might, for instance, involve full-scale management reviews of progress, cost, and future scope. We see distinct advantages in arranging reviews in this way for some projects. On the other hand, small projects that are proceeding along lines of planned scope need not imply major management involvement.

The timing and depth of prototype-end review points needs to be related to the scale and complexity of the system

USE APPROPRIATE MANAGEMENT AND SYSTEM BUILDING TOOLS

Project management tools should support the increased rate of development. It is clearly undesirable if the pace of development is retarded by management review procedures. Likewise, quality review should also keep pace. Ideally, both management and quality reviews should proceed *continuously* with design and construction, with the end date of each prototype version equating to the planned date for the *end*, rather than the *start*, of each phase-end review. Attaining this ideal serves to 'balance up' the project again, in a similar way to the balancing of resources that we discussed earlier on page 14.

Some system building tools allow more continuous review of project deliverables — and thereby help to remove progress bottlenecks

Some system building tools are better than others in allowing a more continuous review of design and other deliverables. We refer here to the various forms of analysis and design aids, data and design dictionary systems, and so on. In the context of this paper, these tools can be considered to be part of the management toolkit.

Such tools support both the activities of the user-designer team, and those involved in quality and management review procedures. One of the features of these tools is that they effectively introduce a degree of separation between design and construction activities. They do this by maintaining (sometimes forcing) separate design deliverables that are extricated from the construction activity. This means that design deliverables can be used (and re-used) independently of other deliverables. The information that can be made available in this way need not be confined to design data. It can include many other project-related items, such as plans and progress reviews.

As far as managing contemporary method development is concerned, there are two key points to look out for in the use of these tools. The first concerns improvements in quality, speed, and efficiency that the tools bring to the design process itself. Because of these benefits, scarce resources are used to maximum advantage, and timescales are shortened by increasing the pace of design work. As a result, the balance between design and construction may be redressed to the advantage of the project.

The second point concerns concurrent working. Even if the speed of design is not increased for the individual there can still be benefits for the team. For example, some system building tools allow developers to have continuous and concurrent access to project deliverables and documentation. This in turn can help reduce the delays incurred at review points where paper documentation otherwise may need to be compiled and circulated for review and discussion.

ACHIEVE BETTER PORTFOLIO MANAGEMENT

More interchangeable resources should mean less lost and unproductive time

Smaller teams, and more interchangeable resources, should mean that it is easier to plan and allocate resources across projects. There should also be a reduction in interference between projects, in terms of time lost in waiting for critical resources, unproductive gaps, and so on. The forward development plan should be less dependent on critical key resources, and more resources should become available because of the reduced maintenance load. Overall, the number of development staff should be much reduced.

Most of these benefits have one thing in common — they will not materialise unless they are planned. Planning tools exist that are designed to help portfolio management. They track project timing and resource requirements, and monitor aggregate needs. In the course of any transition from traditional to contemporary methods, these same tools can be used to plan a number of other things as well, such as retraining needs, staffing levels, critical and peak user involvement, and planned reductions in maintenance effort.

CONSOLIDATE THE KNOWLEDGE GAINED IN USING CONTEMPORARY METHODS

Many organisations are in a period of transition and are experimenting with the use of various forms of contemporary system development methods. As is the case with all emergent techniques offering benefits of time and cost, superior knowledge and expertise can buy competitive advantage.

To achieve the benefits revealed by our survey, however, involves both a financial expenditure and a willingness to learn. Informed knowledge of the effectiveness of a set of tools is required, and this can only be done through objective and consistent measurement of productivity, timescale, and cost. The measurement techniques must allow both for technology factors and for the effects of management decisions regarding timescale and staffing policy. There are changing balances between the various technical and human resource factors, and the measurement techniques must track these trends. Losing sight of the non-technical effects will only obscure the value of otherwise-careful measurement.

Objective, consistent measurement is needed to monitor the changing paybacks from methods and tools

In this area our prescription remains the same: use a discerning metrics programme with a sound quantitative basis. Continuing measurement is needed to monitor both the payback and changes in development performance, in order to feed back into the estimating and planning process. A number of tools are designed to allow both the capture of completed project data, to form an experience database, and also to tune estimating models on the basis of such data.

We have made suggestions for improvement in a number of areas, and in the next chapter, we return to a review of what the benefits should be.

BUTLER COX

Chapter 4

The benefits from better project management

Many of the benefits gained from contemporary development methods derive directly from the use of advanced methods and tools. Better project management should ensure that those benefits are not eroded by poor planning and control.

It is very difficult to measure the potential erosion and hence the benefits of good project management because of all the other environmental factors affecting system development productivity. Nevertheless, an analysis of data from project assessments in the PEP database is consistent with higher productivity being linked to the use of structured methods and more formal project management. We present the evidence below.

It is also possible to identify other business benefits that can be gained by better managing projects that employ contemporary development methods.

THE IMPACT OF FORMAL PROJECT MANAGEMENT ON DEVELOPMENT PRODUCTIVITY

We used the PEP database to try to quantify the effects of using management methods and tools. We asked the PEP sponsors responsible for a sample of completed projects in the database to specify which (if any) formal management methods and tools were used for each project. Our sample was limited to those projects that had used prototyping together with a fourth-generation language to make the comparison as valid as possible.

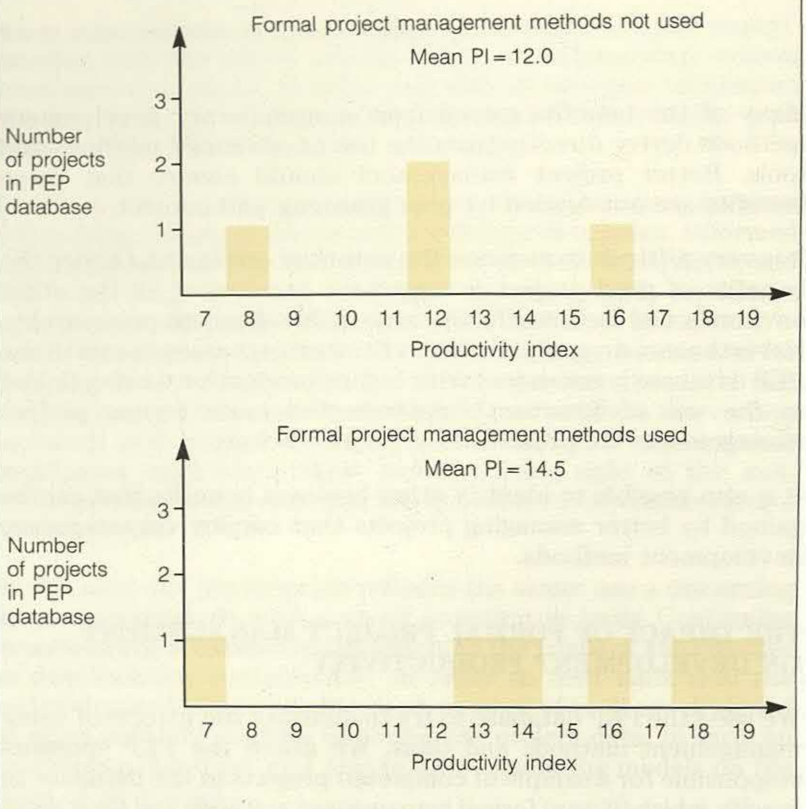
We then compared the productivity index (PI) values of those prototyping projects where formal management methods and tools had, and had not, been employed. (See Figure 4.1.)

The average PI for projects using formal management methods and tools is about two points higher than the average PI for those projects not using such methods.

This evidence must be viewed with caution. Firstly, the sample is too small to prove there is a strong correlation between the use of more formal management and improved productivity. Secondly, even if there were a strong correlation, it would not prove that more formal management caused the improvement. It could be, for example, that certain development teams who have high productivity for some reason, also happen to be ones who favour the use of formal planning methods. It is, however, encouraging to find a difference in PI in the right direction. An improvement in PI by two points can lead to substantial cost savings. (See PEP Paper 5 for examples of the financial value of such savings.) (A similar comparison for projects that had used traditional development methods also showed that those projects for which formal management methods had been used had an

Figure 4.1 Formal project management and productivity

Productivity tends to be higher when formal project management is used.



average PI higher than the rest. The difference in average PI in that case was about four points.)

Analysed PEP data is at least consistent with the view that management methods and tools help improve productivity

BETTER VALUE FOR MONEY

Besides the benefits generated at the individual project level, better project management of contemporary system development should also lead to benefits for the system development function as a whole.

There are several reasons why this can be so: better products, more effective use of resources, and more effort applied to new applications.

There should be better quality products, better use of resources, and better directed development

Better quality products: Nearly all the developers we interviewed believed that with contemporary methods, delivered systems were superior in terms both of more acceptable (to the user) function, and quality.

Better use of resources: Less differentiation in the personal skills required and hence more interchangeability of people allows more efficient use of development resources. Reduced training needs are also important. Most users of fourth-generation languages said that, typically, expertise was acquired in less than a quarter of the time required for traditional languages. Clearly this makes for

BUTLER COX

more effective use of development resources, and also expands the spectrum of skill levels available for use in the system development function.

More new development: In addition, better quality systems and the higher user functionality delivered reduce the system maintenance needed. For a given level of development resources, more effort can therefore be devoted to new development so helping to reduce the application backlog.

FOCUS ON BUSINESS NEEDS RATHER THAN SYSTEM CONSTRUCTION

Using traditional methods, the construction phase of system development was often the main focus of activity and of management attention. The time, effort, and numbers of people involved naturally made this so to the extent that other improvements, at either end of the development cycle (in design and implementation), evolved only slowly. Advances in system building tools have now changed the balance: the *middle stages* of development have shrunk in importance.

Increasingly, development managers can focus more on either end of the development cycle. They can pay more attention to:

The middle stages of development have shrunk in importance

— The management of design, at every level from enterprise modelling to specific requirements definition.

— Implementation management, which has grown in importance because systems are more strategically important and have more widespread impact on the day-to-day operation of the business.

Both ends of the cycle can present complex management problems and are a more fruitful focus for project management. They deal directly with those aspects of applications that are of more direct relevance to the business use of systems.

BETTER RISK MANAGEMENT — MORE FLEXIBILITY

In addition, the normal aim of traditional development has been the creation of reliable, rigidly structured, systems. This aim has been a source of strength but also a weakness. Business requirements change, or are often not clear, and systems that are flexible and can be readily changed can meet business needs more closely. Flexibility was never an explicit aim of traditional methods.

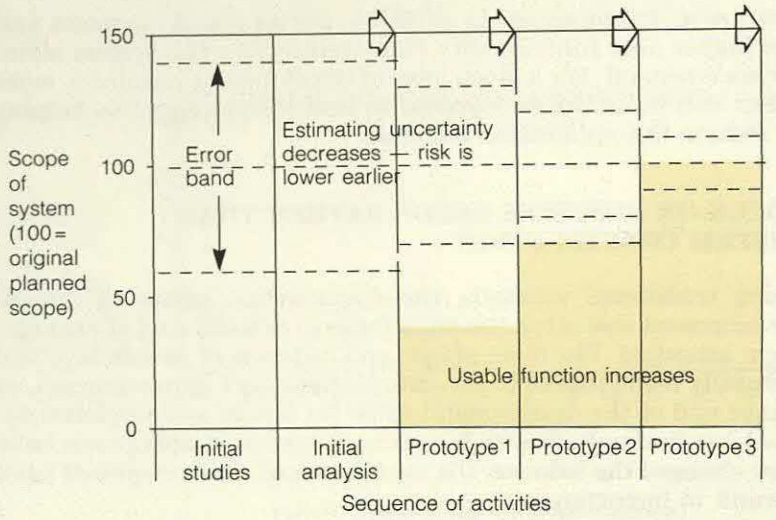
Newer system building tools and methods allow a more flexible approach — more akin to the way business needs and priorities develop

Now, system building methods and tools do allow a more flexible approach, which is more akin to the way in which business needs and priorities develop. Also, evolutionary development, in particular, can afford a fundamentally better approach to risk management.

As we argued earlier, successive prototype versions provide the best phase-end review points for an evolutionary management method, though this will depend on the scale and complexity of the system. Figure 4.2 illustrates how the characteristics of contemporary method development may be exploited to reduce the commercial risks faced in implementing the system and to give better value for money in terms of function delivered for a given investment. Most aspects of the approach apply both to throwaway and evolutionary prototyping, but there are differences of degree.

Figure 4.2 Risk management using prototypes

Prototyping provides better risk management.



Suppose the main project review points are positioned at each major version end-point for which there is a preset functional scope and planned end-date. Then management may rationally and regularly review the following factors:

— Progress against plan, with clear decision points.

— Forecast cost and timescale to completion, on an increasingly accurate basis.

— System functionality to be delivered, progressively amended as business needs, and the cost of meeting them, become clearer.

With regard to the last factor, at each review point the remaining features can be subject to cost-benefit evaluation on a marginal-return basis. This can be both more rigorous and more accurate (in cost-benefit terms) than the *en bloc* approach to such evaluation that is characteristic of traditional development. (With traditional development, typically, all of the features to be built are decided on before construction commences.) An important part of the improved cost-benefit evaluation will derive from clearer user perception (through prototyping) of the effectiveness and value of the features originally requested.

Evolutionary development can provide a better approach to risk management

This step-wise progression is inherently a better risk management system, as far as securing return-on-investment is concerned. It does, however, crucially depend on the system building tools used — if they do *not* permit such an incremental, step-wise approach, then severe penalties (which may be hidden over a period of time) will be incurred.

In any system requirements outline, there will typically be some features that are essential, some that are desirable, and some that are doubtful, or of only cosmetic value. With the evolutionary approach, development can be front-loaded with those features that have the most certain payback.

There are also certain fail-safe aspects to the approach. Step-by-step development permits maximum flexibility in the allocation of effort and use of resources. It maximises the useable development investment at any point in time assuming that decisions to curtail, but not abandon, development are made. Alternatively it minimises the likely wasted investment, in the event that development is abandoned. Useable development effort is likely to be maximised because of the front-loading. In contrast to the traditional approach, there will have been less unproductive analysis and design work in relation to any particular proportion of scope curtailment. With evolutionary prototyping, at any given point of scope curtailment, more effort will also have gone into the functionality built into a useable system.

The benefits of such a prototyping approach are especially important if there are rapidly changing business requirements. Then the allocation of scarce system building resources (of any kind) becomes very important, particularly where there may be a high degree of uncertainty as to the exact nature or value of system requirements.

Development can be more opportunistic both at the project and portfolio level

There is also clearly a potential benefit from using contemporary methods in managing the development portfolio as a whole. Just as methods allow a more flexible allocation of resources and redefinition of system functionality within a project, so they allow more flexibility in allocating resources across the application portfolio as a whole. A well controlled evolutionary approach allows managers to be more opportunistic in developing new systems than when development is based on rigid system development plans and development methods.

CONCLUSION

With contemporary development methods the need for a formal management approach is, in principle, unchanged. However, the development method is the foundation for the management approach — and modifications are required to both, to allow for a number of important differences between traditional and contemporary methods. Unmodified development or management methods will cause problems.

There are very large savings in time and cost that derive directly from the use of contemporary methods and tools and these have important implications, not only for the project but for the organisation as a whole. But to achieve those benefits, the methods used to manage the development projects and supporting tools must be suited to the new methods.

Two aspects of development best characterise the differences between traditional and contemporary methods. Firstly, contemporary methods allow greater flexibility in development (notwithstanding the need for formal management) and offer the potential of a better match between the resulting system and business needs. Secondly, they are bringing about a shift in management focus. Construction, the middle ground of system development, has shrunk in importance, and advances in tools now allow management to focus on those areas of activity that have more direct relevance to the business. The proper task of project management is now, much more, the management of design and implementation effort.

Appendix

Methods and tools for managing contemporary development

Proprietary system development methods and tools can apply either to the development of the system or the management of the development processes but some methods and tools are capable of supporting both.

Most of the methods have been designed to suit the traditional development process but some have been modified to suit contemporary development.

Table A.1 lists some popular development and management methods and shows how far, at the time of compiling the table, they had been modified to suit contemporary development methods.

Table A.2 lists some of the more popular tools used to support project management, showing their applicability to estimating, planning, and control of projects. (We have not included packages devoted only to planning since there are well over a hundred and they have been adequately described elsewhere.)

Table A.1 Some frequently used development and management methods

Type of method			Has it been modified specifically for prototyping?	Supplier
Management method	**Systems development method**	**Combined method**		
		Method/1	✓	Arthur Andersen
		MODUS	In part	BIS
	LSDM			LBMS
PROMPT				LBMS
		STRADIS	✓	McDonnel Douglas
		PRISM	✓	Hoskyns

Table A.2 Some frequently used project management tools

Product and (supplier)	Estimating, planning, and control features								Basis and metrics
	Estimating and sizing				Automatic plan generation				
	Research database	Calibrates to user data	Time-effort trade-off modelling	Portfolio modelling function	Uses SDM template	Links to estimating tool	Exports output to planning packages	Linked planning and control package	
CA-Estimacs (Computer Associates)	✓	✓	✓	✓					Mathematical model High level functionality. Function points.
CA-Planmacs (Computer) Associates					✓	✓	✓	Super Project Plus	
SLIM (QSM)	✓	✓	✓						Mathematical model. Lines of code. Function points. PI, MBI measures.
PMS Bridge (Hoskyns)		User creates estimates			✓ (own)	✓	✓	PMW	Function point-based (manual)
Life Cycle Manager (NASTEC)		User creates estimates			✓ (own planned)	✓	✓	PMW	Task-based (manual)
BIS Estimater (BIS Applied Systems)	✓ (own)	✓			✓ (own)		✓	PMW	Heuristic rules. Logical outputs. Project histories. Proprietary method standards.
Before You Leap (Strategic Systems Technology)	✓	✓	✓		✓ (Cocomo)		✓	Microsoft PMW (dBase Symphony)	Mathematical Model (Cocomo) Lines of code. Function points.

Butler Cox

Butler Cox is an independent international consulting group specialising in the application of information technology within commerce, industry and government.

The company offers a unique blend of high-level commercial perspective and in-depth technical expertise: a capability which in recent years has been put to the service of many of the world's largest and most successful organisations.

The services provided include:

Consulting for Users
Guiding and giving practical support to organisations trying to exploit technology effectively and sensibly.

Consulting for Suppliers
Guiding suppliers towards market opportunities and their exploitation.

The Butler Cox Foundation
Keeping major organisations abreast of developments and their implications.

Multiclient Studies
Surveying markets, their driving forces and potential future.

Public Reports
Analysing trends and experience in specific areas of widespread concern.

PEP

The Butler Cox Productivity Enhancement Programme (PEP) is a participative service whose goal is to improve productivity in application system development.

It provides practical help to system development managers and identifies the specific problems that prevent them from using their development resources effectively. At the same time, the programme keeps these managers abreast of the latest thinking and experience of experts and practitioners in the field.

The programme consists of individual guidance for each subscriber in the form of a productivity assessment, and also publications and forum meetings common to all subscribers.

Productivity Assessment
Each subscribing organisation receives a confidential management assessment of its system development productivity. The assessment is based on a comparison of key development data from selected subscriber projects against a large comprehensive database. It is presented in a detailed report and subscribers are briefed at a meeting with Butler Cox specialists.

PEP Papers
Four PEP papers are produced each year. They focus on specific aspects of system development productivity and offer practical advice based on recent research and experience.

Meetings
Each quarterly PEP forum meeting and annual symposium focuses on the issues highlighted in the PEP papers, and permits deep consideration of the topics. They enable participants to exchange experience and views with managers from other subscriber organisations.

Topics in 1988

Each year PEP will focus on four topics directly relating to improving systems development and productivity. The topics will be selected to reflect the concerns of the subscribers while maintaining a balance between management and technical issues.

The topics to be covered in 1988 are:

- Managing productivity in systems development.
- Managing contemporary system development methods.
- Staffing issues in systems development.
- Managing the maintenance mountain.

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

Belgium and the Netherlands
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
☎ (1) 48.20.61.64, Télécopieur (1) 48.20.72.58

Germany (FR)
Butler Cox GmbH
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) 891 8188

Australia/New Zealand
Mr J Cooper
Butler Cox Foundation
3rd Floor, 275 George Street, Sydney 2000, Australia
☎ (02) 236 6161, Fax (02) 236 6199

Ireland
SD Consulting
72 Merrion Square, Dublin 2, Ireland
☎ (01) 766088/762501, Telex 31077 EI,
Fax (01) 767945

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

The Nordic Region
Statskonsult AB
Stora Varvsgatan 1, 21120 Malmo, Sweden
☎ (040) 1030 40, Telex 12754 SINTABS

Spain
Associated Management Consultants Spain SA
Rosalía de Castro, 84-2°D, 28035 Madrid, Spain
☎ (91) 723 0995