

# CINDERELLA GETS AN INVITATION: RECOGNISING AND ORGANISING SYSTEMS MAINTENANCE & SUPPORT

C. James Bacon.

Department of Information Systems  
University of Canterbury, Christchurch, New Zealand  
JamesB@Accy.Canterbury.ac.nz

## ABSTRACT

Systems maintenance of existing systems is a 'Cinderella' subject in the field of information systems & technology (IST). Although it accounts for the majority of systems work, the subject is generally neglected and taken for granted. Based on studies carried out in a large Australasian organisation, evidence is presented for recognising the strategic role of systems maintenance, as a complement to new development. Alternative means of organising systems development work is examined, with the conclusion that a separate systems support group for carrying out maintenance may have net advantage. A re-definition of maintenance, according to a taxonomy of eight different types, is proposed. Effective methodology and metrics for maintenance are also addressed, together with the general phenomenon of 'entropological deterioration', and 12 factors that need to be used in assessing existing systems.

## INTRODUCTION

The subject of systems or software maintenance is a 'Cinderella' in the field of information systems & technology (IST). In the academic environment, teaching is oriented almost exclusively to new systems development. What happens to those new systems after they are installed is of little concern. Also, compared to the plethora of textbooks on new development, there would be few books suitable for a course on systems maintenance. Similarly, it is a relatively neglected subject in the journal literature on IST.

*In the business environment, managerial focus tends to be on new development, with little recognition of systems or software maintenance as a key activity. As for systems developers, maintenance tends to be something you do when you get the time, or delegate to trainees. It might even be contracted out; anything so that you don't have to do the 'dirty work' yourself. And software tools, notably CASE tools, concentrate on new development.*

*Thus, just like Cinderella in the children's story, systems maintenance is generally neglected, abused and taken for granted.*

Systems maintenance is the process of modifying existing, operational systems. It is referred to alternatively as *software* maintenance, but is herein referred to as *systems* maintenance. In most organisations (apart from vendors of systems software), it applies largely to application software rather than systems software, and it incorporates not only the software but the documentation and procedures, overall design, and hardware platform as well, i.e. the whole system. Also, in terms of the organisational activity, there is some evidence that "maintenance" is now being replaced by the term "support" in referring to the activity of modifying systems. Therefore, "systems support" may become the more prevalent term, rather than systems maintenance, at least in referring to the function that carries out systems maintenance.

Returning to the storybook parallel, there are a number of reasons why Cinderella needs to receive an invitation to the party, and get the recognition she deserves from the people at the top.

First, various estimates indicate that between 50 per cent and 80 per cent of systems work is taken up with systems maintenance and support (Gibson and Senn, 1989 and Vail, 1991). Systems maintenance therefore represents the majority of systems work. Related to this point is that, as time goes on, systems are becoming bigger, growing at about ten per cent per year (Harrison and Cook, 1990). This growth may accelerate, to the extent that systems become oriented to graphics, and the needs of human-computer interface. As systems become bigger, they generally become more complex. Consequently, effective systems support is becoming more critical, and a major requirement, in many organisations.

A more fundamental reason why systems maintenance and support needs to receive better recognition and management is that, although they are not normally represented as such in annual reports, existing information systems represent a significant corporate asset. For most organisations, especially those of any size, day-to-day survival and excellent customer service can depend upon the

organisation's information systems. Therefore, somewhat like a building but more so, a business application system will quickly deteriorate and lose value if it is not properly maintained and managed.

In 1992, the author was invited to undertake an organisational review of the systems development and maintenance activities in a large Australasian organisation operating in the services sector. The basic need was to enhance the effectiveness of the organisation's IST function. This was especially needful at a time when the business sector concerned was undergoing rapid change and an intensifying level of competition. The organisation therefore needed full and effective support from IST. The review was an opportunity to study practice, and apply as well as validate relevant research findings. It represented a response to the call for greater research investigation in the field (Computer Science and Technology Board, 1990). In the event, it represented a case study in recognising and organising systems maintenance and support.

### THE ROLE OF SYSTEMS MAINTENANCE & SUPPORT

The basic role of systems maintenance & support is indicated in a proposed Systems Service-Product Grid, as shown in Figure 1. This is based on the Strategic Grid (Cash, McFarlan and McKenney, 1988).

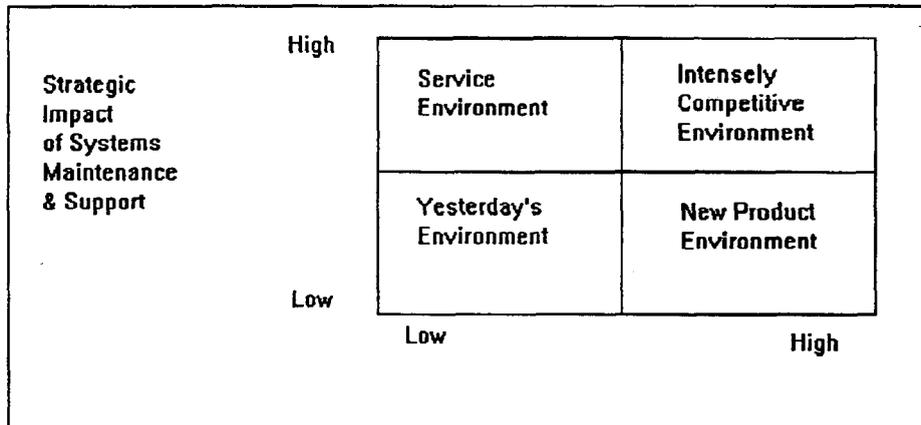


Figure 1. The System Service-Product Grid

Where the strategic impact of systems maintenance and support is high, there is a service environment. That is, the organisation is not just dependent on the existing installed systems for its day to day operations and survival. It is also dependent on its users receiving effective service, in responding to their continually changing needs with those systems. Thus, a more appropriate term for users, in this context, is customers or clients (the term "clients" denoting a somewhat closer business relationship than "customers"). This dependency is a characteristic of the situation where excellent service to the organisation's external customers is a critical success factor. For that service to be excellent, users/clients are dependent, directly or indirectly on the service they receive from the systems support function. Implicit in the continually changing needs of users/clients is the need for continuous improvement. This is a fundamental concept in total quality management (TQM). Thus, evolutionary or incremental improvement, in small steps, is important to systems quality, and a key to excellent service in systems support.

In contrast, where there is a continual and strategic need for new systems development, there is a new (systems) product environment. This is likely to be the case where the business sector or type of business is obliged to engage in a continuous process of bringing out new or differentiated business products and/or services. It also occurs where 'orders of magnitude' improvement in business functions are the focus, as in business process re-engineering. A third case is where the organisation needs to be at the 'leading edge', in bringing in new information systems to implement new information technology.

If systems maintenance and new systems development are both strategically important, then the organisation is operating in an intensely competitive environment. However, where maintenance and new development both have a low impact on the organisation, it is evident that the organisation

is not utilising information systems & technology as a strategic resource, and is operating in 'yesterday's' environment.

The implication is that both systems maintenance and new systems development need to be aligned with organisational goals. Both activities need to be synchronised with the organisation's overall strategy (Swanson and Beath, 1989).

Therefore, the primary task of the organisation's IST function may not be new development, nor systems maintenance. Rather the concept of a *dual* task may be more appropriate. This implies that when efforts are directed at reducing the size and expense of the maintenance task, the motivation may be wrong, if it is to free up resources for new development. The implication in this case is that maintenance is inherently a secondary consideration, with limited strategic impact.

The limited role in which maintenance is traditionally viewed is largely based on the traditional lifecycle, as represented in the 'waterfall' model (Boehm, 1981). That is, maintenance is 'tacked on' at the end of new development, almost as an afterthought.

The alternative view is that of maintenance integrated into the whole lifecycle. Systems design includes an orientation to effective maintenance, and there is a methodology of/for maintenance.

The alternative is also a wider view of existing systems as a portfolio of strategic assets. Under this view, the systems support function has a key responsibility in maintaining and protecting the portfolio in good order (Swanson and Beath, *ibid.*, and Hayes, 1991). This is reported to be the view in 'leading edge' organisations (Layzell and Macauley, 1990). In such organisations, the role of systems support is now recognised as being of key importance in the overall IST strategy.

It has been said that the major challenge for software engineering in the 1990s is systems maintenance (Corbi, 1989). To the degree that this is true, systems maintenance & support needs to be properly recognised, properly organised, and properly managed. This applies not only to 'leading edge' organisations, but any organisation that aims to use IST effectively in an increasingly competitive environment.

#### CHALLENGES INVOLVED IN THE COMMON FORMS OF SYSTEMS ORGANISATION

In the organisation that was the subject of the organisational review, the division of work among its systems staff was, first, according to business application area, and second, by type of work. Staff specialised in specific business domains and, within these domains, there were analysts and programmers. This has been described as an Application-Work or A-W type of systems organisation, which is a common form of organising systems staff (Swanson and Beath, 1990).

The 'pure' W type of organisation is the more traditional type of organisation, wherein programming expertise is required in third or second generation languages, and separated from analyst work. There is no specialisation by business domain. The contrary form of specialisation by business domain (type A), is more common in today's organisation, since it allows analyst-programmers (both types of work combined in a single type of position), to develop specialised application and business area knowledge, as well as closer relationships with a given user clientele. Thus, in the organisation concerned, the application (A) and work (W) forms of organisation were combined in hybrid form, which is not unusual. A third form of organisation is the lifecycle or L type organisation (Swanson and Beath, *ibid.*). In this case, analyst-programmers specialise in either new development or systems support (maintenance).

In order to enhance the effectiveness of the subject organisation's IST function, it was recommended, and it was agreed after discussion, that it should move to an L type of organisation, thus separating maintenance work from new development. There were a number of reasons for this, based on the challenges found with present structures.

One of the challenges faced was the inherent lack of flexibility in the A form. There were 'peaks and valleys' in the need for new development in each business application area. Systems development needs were 'lumpy'. This meant that systems staff could be overworked in one period while waiting for work in another. It also meant that systems staff could be overworked in one application area while waiting for work in another. The overall business impact was that there was a frequent short-fall of resources needed to support business needs, together with user concerns about systems expense and lack of support from the IST function. Thus, there was a need for greater flexibility and response in covering the different application areas.

Another challenge faced by the organisation was that users/clients were not happy with the level of service they were receiving, on seemingly simple change requests, on current systems. In some cases, the response to such requests was so long in coming that the user/client had given up on and

forgotten the request, with the change having been superseded by others. Thus, there was a need to enhance the level of service in response to change requests on current systems.

Generally, management of maintenance was limited. Statistics were kept on the amount of "re-work" or corrective maintenance being done. However, they had limited use, since there was no specific programme or response for preventive measures. Maintenance was seen as being equivalent to re-work, made necessary by new development not being done properly. Maintenance was therefore seen in very much of a negative light, as in many IST environments. That is, there was a stigma attached to systems maintenance.

## THE PROS AND CONS OF SEPARATING MAINTENANCE FROM NEW DEVELOPMENT

Owing to the negative view attached to systems maintenance in the subject organisation, at least according to the traditional view held at the time, there was resistance to the idea of separating it from new development. No-one wanted to be assigned to a separate maintenance function, having to clean up 'other people's mess'. The specific reasons initially given against the recommendation to separate maintenance from new development were as indicated in Figure 2. These reasons constitute the 'cons' of separating maintenance from new development.

- |  |
|--|
| <ol style="list-style-type: none"><li>1. Separate maintenance staff may not understand a system and the effects of requested changes as well as those who developed it, so that systems quality may suffer.</li><li>2. Separate maintenance people would be correcting other people's mistakes, and it would be better if people corrected their own.</li><li>3. Response to users/clients might be slower, since separate maintenance staff may not be as familiar with the specific business/area concerned, and would have to study and be dependent on systems documentation more than those who developed a given system.</li><li>4. Communication costs would be higher, since there would be a need for maintenance staff to refer to the new development staff concerned, and there would be duplicate interface with users/clients.</li></ol> |
|--|

Figure 2. The Cons of Separating Maintenance from New Development

There are advantages and disadvantages in most organisational forms. Based on observation and study in the subject organisation, advantages or 'pros' were advanced in favour of a separate systems support (maintenance) group, as indicated in Figure 3.

The organisation concerned did not produce any tangible product, but as a service organisation was in the 'information business'. It was a veritable 'information factory'. Its information systems therefore represented its 'productive equipment' and the engine that 'powered' its operations.

The portfolio view of installed application systems was therefore of particular relevance to the organisation concerned, and this concept was emphasised in arguing the case for effective maintenance, and a separate maintenance function. However, in moving to a separate maintenance function, management felt it was important to distance itself as far as possible from the maintenance 'stigma', by avoiding the word "maintenance" in naming the new group. It was therefore given a more client-centred name, focusing on rapid service and quality support.

A separate systems support group also addressed the need for greater flexibility in allocating new systems development staff across different business domains. Essentially, it was recognised that systems staff needed to become cross-functional professionals, able to operate in different business application areas. In this way, project teams could be pulled together for a given new development project, which would mean more effective response to and support for the organisation's business needs. To make this happen, all staff were expected to do a 'tour of duty' in the systems support group, where they would be exposed to the maintenance needs of several business areas, thus obtaining 'hands on' training across different application areas.

It was important that such 'hands on' experience in systems support then became a basis for further promotion. It was also important that systems support was be staffed by experienced people. It was not to be used as a training ground for beginners. Implicit in this was the implication, and the 'signal' from management, that maintenance was in no way secondary to new development work.

1. There is reduced risk exposure for the organisation, in terms of: (a) loss of knowledge when a person leaves, and (b) lack of availability in the event of absence due to various reasons, when at least two people (a new development person and a systems support person), have knowledge about a given application area.
2. System documentation is forced to improve when people other than those who originally developed a system are maintaining it.
3. There is reduced interference with and distraction from new development, thus enhancing the prospect of meeting project schedule needs.
4. Since effective maintenance is the focus in systems support, rather than new development, 'quick and dirty' fixes, or 'patches', tend to be avoided, and the design integrity and maintainability of installed systems is enhanced.
5. The principle of continuous, incremental improvement in the quality of existing systems, as per total quality management (TQM), receives enhanced focus.
6. Ongoing response to users/clients becomes a primary focus, with positive implications for user/client satisfaction, since maintenance change requests are not seen as a distraction and a 'nuisance' interfering with new development, but are instead the main job.
7. Specific responsibility can be assigned for 'custodianship' of the organisation's installed systems, within a systems support group, in accordance with the asset/portfolio view.
8. Effective management of systems maintenance becomes a specific responsibility, and effective methodology and metrics become more feasible.

Figure 3. The Pros of Separating Maintenance from New Development

### RE-DEFINING MAINTENANCE

A fundamental reason for the commonly negative view of systems maintenance is its traditional orientation to corrective work, and this at the coding level rather than the design level. In the subject organisation, maintenance was generally known as "re-work", i.e. corrective maintenance. But this is only one kind of maintenance need and, from a strategic perspective, not the most important.

Three kinds of maintenance have been defined: corrective, adaptive and perfective (Lientz and Swanson, 1978). These three kinds have wide acceptance at the present time, in representing the totality of systems maintenance work. Corrective maintenance is defined as that carried out to repair or correct processing performance and implemented failures; adaptive maintenance responds to required changes in the data and processing environments; perfective maintenance eliminates processing inefficiencies and enhances performance, (Swanson and Beath, 1989).

But other types of maintenance have also been defined. For example, there is functional maintenance, which specifically refers to minor changes to or enhancements of system functions, (Hall, 1987). Two further kinds of maintenance that have been proposed are: (1) preventive maintenance, to improve future maintainability and reliability, and (2) deletion maintenance, to delete capabilities that are no longer needed (Computer Science and Technology Board, *ibid.*). Thus, there have been at least three others added to the three kinds of maintenance commonly seen as constituting systems maintenance work. In addition, two further kinds or types of maintenance are herein proposed, namely mandated maintenance and post-implementation maintenance. The different kinds are indicated in Figure 4, which represents a taxonomy of systems maintenance types. The taxonomy re-labels deletion maintenance as rationalisation maintenance, and includes definitional enhancement of the other types. It also distinguishes between discretionary and non-discretionary maintenance.

If systems maintenance work is to be properly recognised, organised and managed, the specific kinds of maintenance need to be identified and articulated. The essential reason for this is 'granularity', in which respect there is a parallel with project management. If a systems development project is not divided into discrete phases, activities and tasks, planning and control cannot be effective. Also, metrics maintained on such projects are of limited value, since there can be no meaningful validation (Jones, 1991). Thus, effective management of systems maintenance needs to include, specifically, planning and control of the different kinds of maintenance required. Distinction between the different kinds is also needed because the orientation and response to each needs to be different.

<u>Non-Discretionary Maintenance</u>	
1.	Corrective Maintenance: the repair or correction of design errors, including emergency changes
2.	Adaptive Maintenance: modification made necessary by changes in the processing, data base or network environments.
3.	Mandated maintenance: modification made necessary by changes in the organisation's external environment, as in legislative and tax changes.
<u>Discretionary Maintenance</u>	
4.	Perfective Maintenance: the enhancement of processing, data base, or network efficiency.
5.	Rationalisation Maintenance: the deletion of superfluous output, input, programs, data, or redundant processing.
6.	Preventive Maintenance: design and documentation changes for the purpose of improving future maintainability and/or reliability.
7.	Functional Maintenance: minor additions to or enhancements of system functions, in response to evolving user/client needs.
8.	Post-Implementation Maintenance: the addition or enhancement of functions following system implementation, as agreed during the process of new development.

Figure 4. A Taxonomy of Systems Maintenance Types

Efforts to reduce the amount of maintenance work need to focus on non-discretionary maintenance, since this is the form of systems maintenance that the organisation is forced to do. In the case of mandated maintenance, for example, this could be combined with types of discretionary maintenance, so that the time and effort expended obtains some value. Also, rationalisation maintenance reduces system size and complexity, by getting rid of 'petrified code' (Carlyle, 1989). Consequently, it helps to reduce corrective maintenance, and makes other kinds of maintenance easier. Preventive maintenance can have a similar effect on non-discretionary maintenance.

Discretionary maintenance is the form of systems maintenance which the organisation chooses to do, and for which it needs to plan on a continuous basis. For example, if project target dates are to be met, it is necessary to plan for post-implementation maintenance, as defined. Otherwise, inclusion of functions that users/clients request after project schedules have been agreed can have a negative impact in meeting such target dates. Also, there needs to be planning for functional maintenance, particularly on those application systems which are central to business functions and activities.

Implementation of a taxonomy of systems maintenance types, as a chart of accounts in planning, recording and controlling maintenance work, will provide a clearer picture of maintenance, and facilitate effective management. For example, it will facilitate meaningful metrics, as in the proportion of resources consumed by discretionary as opposed to the non-discretionary form of maintenance, and in the different types of maintenance. The taxonomy may also help to remove the traditionally narrow view of systems maintenance.

The taxonomy was therefore recommended, and agreed, as a means of improving overall management of systems maintenance in the subject organisation. It was screened and reviewed by experienced, senior-level analysts and programmers, and found to be useful as a means of approaching systems maintenance, and as a basis for providing overall measures for the systems support group.

#### MAINTENANCE METHODOLOGY AND DOCUMENTATION

Under the commonly narrow view of systems maintenance, the term "maintenance methodology" may seem academic. Under this view, you: (1) study the source code to understand the problem, (2) make the needed changes to the code, and (3) test the changes, which is hardly a methodology. This perspective and approach is unlikely to be associated with the portfolio view of existing, installed systems.

Under the portfolio view, and as part of an overall maintenance methodology, it would be necessary to periodically assess each installed system, with respect to its strategic, functional and technical value (Hayes, 1991). The assessment would need to include factors such as those listed in Figure 5. The implication in all this is that there may be benefit in some form of impact analysis, prior to coding work. This involves an analysis of the effects of the prospective changes on overall system design.

There are therefore a number of considerations that need to be incorporated in an effective maintenance methodology. These considerations were articulated and discussed in recommending certain enhancements in the approach to systems maintenance in the subject organisation.

<p><u>Strategic Factors</u></p> <ol style="list-style-type: none"><li>1. Pertinence to and/or role in organisational plans.</li><li>2. Effectiveness in supporting business critical success factors.</li><li>3. User/client satisfaction.</li><li>4. Importance and value of the business function concerned.</li></ol> <p><u>Functional Factors</u></p> <ol style="list-style-type: none"><li>5. The dependence of the business function on the system.</li><li>6. Effectiveness in supporting the business function.</li><li>7. Costs of operation and maintenance.</li><li>8. Back-log of change requests.</li><li>9. Utility and use made of respective screen and printed output.</li></ol> <p><u>Technical Factors</u></p> <ol style="list-style-type: none"><li>10. Maintainability (modularity, simplicity, consistency and self-descriptiveness).</li><li>11. Reliability, (as per operations records/logs).</li><li>12. Viability and vendor support.</li></ol>
--

Figure 5. Assessment Factors for an Installed Application System

The assessment factors need to part of a maintenance methodology, and be included at the 'front end'. But maintenance also begins early in new development, rather than at its 'back end' (Pfleeger and Boehner, 1990). Thus, maintenance and the inevitability of future change need to be a consideration throughout new systems development. That is, evolving user/client needs to be explicitly recognised.

A further maintenance methodology consideration is that a system tends to lose its architectural integrity, if maintenance changes are at the code level only, with overall design excluded. This phenomenon might be referred to as entropological deterioration. Fourth generation languages (4GLs), and object orientation may help to reduce this. However, to the degree that they encourage a 'lazy' development methodology, and prototyping as a substitute for (instead of a complement to) formalised methodology, they may exacerbate the phenomenon. With this deterioration the system becomes more difficult to understand, in making changes to it. In the latter respect, it has been estimated that this traditional first step of understanding existing code and documentation consumes 47 to 62 per cent of programmer's/analyst's time (Hayes, *ibid.*).

As part of the recommended changes, the iterative or transformation approach to systems maintenance was recommended, for inclusion in a revised methodology, as an alternative to the extremes of the quick fix and total re-write. This is based on the idea of recovering design abstractions, in re-using design and code components from the existing system (Arango et al, 1986, and Harrison and Cook, *ibid.*). Thus, instead of starting with and focusing on source code, the approach starts with the design of the existing system, starting at the highest or overall level then working down, re-using existing design and code as much as possible. The iterative/transformation approach is not a re-write of the existing system, but it is more than the quick fix approach. It is a middle course between the two. The benefit is that it promotes continuous evolution of the system design, thus avoiding entropological deterioration, and making future maintenance easier. The result of these considerations was a recommended maintenance methodology for the subject organisation.

The portfolio & design-based maintenance methodology implies that there are certain documentation needs. For example, at the very beginning there is a need to document pertinent aims, objectives and critical success factors, and to document the present system and/or procedures. There is also a need to document each maintenance change, or change release, including its

justification, reason and purpose, some of which can be summarised in the source code itself. This type of documentation is often missing in the traditional approach to systems maintenance, which generally provides little more than the latest code (Baxter, 1992). With this approach, there is a virtual need for the original developer(s) to continue personal involvement in the installed system. Thus, effective maintenance documentation is complementary to effective maintenance methodology.

As part of the enhanced maintenance methodology for the subject organisation, systems support staff were to be involved in new systems design. As 'custodians' of the portfolio of installed systems, they were to participate in structured walkthroughs, and in post-implementation maintenance. There was also a formalised Maintenance Handover Phase, wherein systems support staff were to become familiar with the new or enhanced system, and obtain agreed documentation for ongoing maintenance.

## MAINTENANCE METRICS

It is difficult to know how well you are doing in a given area of activity, or the difficulty of what you are trying to do, without appropriate and meaningful measurement. The organisation and management of systems maintenance and support can only, therefore, be deemed effective and successful to the extent that it is measured and validated.

The need for measurement in the area of information systems & technology is directly related to its economic impact (Jones, *ibid.*). The surging interest in software quality measures, project productivity, and IST value-for-money indicators is therefore an indication, if any were needed, that IST has become a strategic economic resource, and a vital area of activity for today's organisation. It is likely that such measures will become increasingly important.

However, there are a number of imperatives in any maintenance measurement programme. First, it needs to be fully supported by senior management, because: (a) it involves cultural change and therefore needs authoritative endorsement, and (b) it requires resources, in a similar way to which cost and management accounting require resources. Second, maintenance measurement requires learning and training. Third, it needs to be positive rather than punitive. In the latter respect, people need to be rewarded for finding errors, rather than being faulted for making them (Robinson, 1993). The fourth imperative is, therefore, that effective measurement and targets need to be built into the reward system for management and staff.

There are four general areas where metrics can be applied to maintenance. These are shown in Figure 6.

- |  |
|--|
| <ol style="list-style-type: none"><li>1. Taxonomical: measure and compare the different kinds of maintenance.</li><li>2. Quality &amp; Maintainability: measure ease or difficulty of maintenance.</li><li>3. Soft Factors: measure human and environmental factors.</li><li>4. Productivity: measure performance:</li></ol> |
|--|

Figure 6. General Areas of/for Maintenance Metrics

The taxonomy and chart of accounts for the different types of maintenance work being done provides a means of comparing the proportion of maintenance effort being taken up by each type or each account. It also facilitates metrics specific to each type

A second area for maintenance metrics is quality and maintainability, although this applies as much to new development as it does to maintenance. For maintenance, systems quality applies specifically to maintainability. The four attributes of systems software and documentation that optimise maintainability are consistency, simplicity, modularity and self-descriptiveness. Consistency refers to the uniformity and standardisation of design, notation and techniques. Simplicity refers to ease of understanding and avoidance of complexity. Modularity refers to a design structure of independent modules. Self-descriptiveness refers to the degree to which the software and documentation provide explanation of functions.

A third area is that of 'soft' factors, which involves measurement of: such things as: (1) the morale of systems support staff, (2) the level of service perceived by clients/users, (3) the cooperation and assistance received from clients/users, (4) the skill, experience and training needs of systems support staff, and (5) environmental assistance/constraints, as in the provision of technical and/or physical resources.

A fourth area is that of productivity. However, although metrics have been developed for measuring new development, there has been very little research in applying productivity metrics to systems maintenance. Function point analysis (FPA), was originally developed for new systems development (Albrecht, 1984). With the increasingly strategic importance of IST, and the need for optimising systems development productivity in supporting business needs, FPA is enjoying increased usage. There is even an active International Function Point Users Group (IFPUG). FPA appears to hold the greatest promise for measuring maintenance productivity. It has had some application, specifically to three kinds of maintenance defined in the taxonomy, namely perfective, functional, and post-implementation maintenance (Cote and St-Pierre, 1990).

From an overall perspective, metrics for maintenance need to focus on those areas that represent or include critical success factors (CSFs) for the systems support group or maintenance activity (as well as for the IST function as a whole). For example, if user/client satisfaction is a CSF, then this needs to be regularly measured, likewise 'turnaround time' for maintenance changes.

In the organisation concerned, metrics were being maintained on the amount of corrective maintenance being performed, and it was recommended that the area(s) of measurement be enlarged, particularly in the areas of maintainability and soft factors. This was agreed, and it was accepted that the systems support group depended, for its effective management, upon the development of a metrics programme suitable to the organisation.

### OPTIMISING USER/CLIENT SERVICE

A fundamental aim and CSF for a systems support group is user/client satisfaction. This will depend to a large extent on the level of service and support received and perceived by the user/client. "Perceived" is the operative word, because systems staff may be exerting great effort to provide good service and support to users/clients, but still the perception could be that of poor service. Two basic reasons for this are: (1) lack of user/client awareness, and (2) 'red tape'. To become aware, users first need to be involved in the maintenance methodology, and be part of if not leading the assessment of existing systems. Second, users need to be involved in the planning and prioritisation decisions. Third, users need to be educated, so that they can participate as partners with systems management and staff in these decisions. A fourth area of needed user awareness is that of change request status. Users/clients need to know whether and when a given change request is scheduled for implementation, its expected release date, and its progress (in the event of enquiry), as it progresses toward implementation.

With respect to 'red tape', the maintenance methodology must not become so bound up that it obfuscates and inhibits communication between users and systems staff. Where necessary, communication needs to be direct, as between the initiator of a change request and the systems support group.

Both awareness and 'red tape' needed to be addressed in the organisation concerned, and recommendations were made and agreed in these areas. The aim was an enhanced level of service and support to users/clients, both in reality and in the perception of users/clients.

### MANAGING AND MOTIVATING CHANGE

Where an organisation is oriented to the traditional view of systems maintenance, a process of cultural change is required, in moving to a strategic asset, portfolio-based view of systems maintenance and support. This involves the management of change, both organisational and behavioral.

The management of change needs to incorporate two basic principles: (1) the participation of those affected by or involved in the change, and (2) an orientation to identifiable results (Mankin et al, 1988, and Schaffer and Thomson, 1992). The principle of participation was embraced in a wide consultation process during the study, involving all levels in developing the recommendations. In addition, the very first recommendation embraced both principles. It recommended a process whereby the recommendations were divided up and allocated among teams of systems staff, with each team being asked to assess the following, for each recommendation allocated to the group: (1) whether the recommendation should be accepted or rejected, (2) whether and where it needed to be refined, modified or improved, (3) whether it needed to be replaced by a better alternative, as specified and justified, (4) other things that needed to be done or considered for effective implementation, (5) additional or re-oriented resources that might be needed for effective

implementation, (6) specific measures or criteria that could be used in determining whether implementation was proceeding successfully, and (7) priority for implementation. The teams were asked to submit a one-page report on each recommendation. Management also decided to establish an overall change process team, representing different levels and areas, to oversee the change process.

## CONCLUSION

The subject of systems maintenance and support is potentially a large subject area. To date, it has been treated as a 'Cinderella' subject. Application systems are growing, becoming more complex, and increasingly represent a strategic asset for most organisations. But poorly maintained systems may represent a 'time bomb' for many organisations. Thus, systems maintenance needs to receive a higher profile, and considered complementary rather than secondary to new systems development. This paper focuses on organising and managing systems maintenance and support, and directs itself to the main areas involved. Not more than a handful of researchers have focused on systems maintenance, and this paper owes a debt to them since it builds on their work. But, given the increasing significance of maintenance, there is scope for further research; a lot more work needs to be done.

## REFERENCES

- Albrech, Alan J. (1984) **Productivity Measurement and Estimate Validation**, Purchase, NY: IBM Corporate Systems & Administration.
- Arango, G., Baxter, I., Freeman, P., and Pidgeon, C. (1986) "TMM: Software Maintenance by Transformation", **IEEE Software**, May, Vol. 3, No. 3, pp 27-39.
- Baxter, Ira D. (1992) "Design Maintenance Systems", **Communications of the ACM**, April, Vol. 35, No. 4, pp 73-81.
- Boehm, Barry W. (1981) **Software Engineering Economics**, Englewood Cliffs NJ: Prentice-Hall.
- Carlyle, Ralph E. (1989) "Fighting Corporate Amnesia", **Datamation**, 1 Feb., Vol. 35, pp 41-44.
- Cash, J.I., McFarlan, F.W. and McKenney, J.L. (1988) **Corporate Information Systems Management: the Issues Facing Senior Management** Homewood, Illinois: Dow Jones-Irwin.
- Computer Science and Technology Board (1990) "Scaling Up: A Research Agenda for Software Engineering", **Communications of the ACM**, March, Vol. 33, No. 3, pp 281-293.
- Corbi, T. (1989) "Program Understanding: Challenge for the 1990s" **IBM Systems Journal**, Vol. 28, No. 2, pp 294-306.
- Cote, Vianney and St-Pierre, Denise (1990) "A Model for Estimating Perfective Software Maintenance", **Conference on Software Maintenance 1990**, pp 328-334. Los Alamitos, CA: IEEE Computer Society Press.
- Gibson, Virginia R. and Senn, James A. (1989) "Systems Structure and Software Maintenance" **Communications of the ACM** March, Vol. 32, No. 3, pp 347-358.
- Hall, Rand P. (1987) "Seven Ways to Cut Software Maintenance Costs" **Datamation**, 15 July, pp 81- 84.
- Harrison, Warren, and Cook, Curtis. (1990) "Insights on Improving the Maintenance Process through Software Measurement" **Conference on Software Maintenance 1990**, pp 37-45. Los Alamitos, CA: IEEE Computer Society Press
- Hayes, Ian (1991) "Software as Assets Increases Returns", **Computerworld New Zealand**, 23 Sep, pp 13-15.
- Jones, Capers (1991) **Applied Software Measurement**, NY: McGraw-Hill
- Layzell, P.J. and Macaulay, L (1990) "An Investigation into Software Maintenance-Perceptions and Practices", **Conference on Software Maintenance 1990**, pp 130-140. Los Alamitos, CA: IEEE Computer Society Press
- Lientz, B.P. and Swanson, E.B. (1978) "Characteristics of Software Maintenance", **Communications of the ACM**, June, Vol. 6, pp 466.. 21, No
- Mankin, D., Bikson, T., Gutek, B. and Stasz, C.(1988) "Managing Technological Change: the Process is Key", **Datamation**, 15 Sep. pp 69-75.

- Pfleeger, S. and Bohner, S (1990) "A Framework for Software Maintenance Metrics" **Conference on Software Maintenance 1990**, pp 320-327. Los Alamitos, CA: IEEE Computer Society Press
- Robinson, Jeffery A. (1993) "The Dark Side of Software Metrics", **Information Strategy: the Executives Journal**, Winter, pp 44-47.
- Schaffer, Rober H. (1992) "Successful Change Programs Begin with Results", **Harvard Business Review**, Jan-Feb., pp 80-89.
- Swanson, E. Burton and Beath, C. (1989) **Maintaining Systems in Organisations**, NY: John Wiley and Sons
- Swanson, E. Burton and Beath, C. (1990) "Departmentalisation in Software Development & Maintenance, **Communications of the ACM**, June, Vol. 33, No. 6, pp 658-667.
- Vail, Simon (1991) "Happiness is a Fault-Free System", **Management Today**, Nov. pp 115-116.