Transactions on Information and Communications Technologies vol 8, © 1994 WIT Press, www.witpress.com, ISSN 1743-3517

Towards a quality management system for end-user application development

S. Barr,[a] R.W. Foley[b] & M. McMullan[b]
[a]*Department of Business Administration*
[b]*Department of Computer Studies, Glasgow Caledonian University, Glasgow, G4 0BA, UK*

ABSTRACT

The advent of the personal computer (PC) has brought many significant changes to the application of computers within the everyday workplace. One particular area of change is in the growth of what is termed End-User Computing (EUC). Here non-computer specialists develop their own applications using commonly available PC-based application packages.

Without realising it, many end-users are developing what computer professionals would term "software". However, unlike professionally developed software, there has been little emphasis on development methodology or quality management. As a result, end-user developed applications have often contained significant errors in reliability and prove difficult to maintain.

With the adoption of Quality Management Systems becoming the norm for the professional computing community, there is now a clear need for end-user developed applications to follow at least, a clear set of similar guidelines, where appropriate. The purpose of this paper is to identify a proposed framework to enable end-users to develop their own software within an overall quality environment. Clearly, such guidelines must be sufficiently rigorous to ensure confidence in the developed software but must not be seen as overly bureaucratic and hence burdensome within a non-computing professional context.

INTRODUCTION

Over the last 10 years or so the role of the end-user in computing has changed quite dramatically. Prior to this period, only a minority of managers and

professionals would have used computers in any significant aspect of their work. Those who did, would normally make use of large systems which had been developed by the organisation's Data Processing (or Information Systems) department. These systems would operate on a mainframe computer and access to them was provided via a simple terminal. The use of the computer would only form a small portion of their overall job.

Since the launch of the personal computer, this picture of computer usage has changed quite dramatically. Today, most managers and professionals have integrated the computer into their mainstream work, indeed many could not function satisfactorily without it. This has caused a radical decentralisation of applications development with end-users themselves creating and tailoring their own business applications using a wide variety of PC-based packages. This trend has become known as end-user computing (Panko[1]).

Some of these end-user developed applications have been used in critical areas of individual department's work, and indeed a significant proportion have involved multiple functions and departments (Rockhart & Flannery[2]). One aspect of major concern is that very little emphasis has been placed on, what computer professionals would consider to be their "bread-and-butter", development methodology (Bray[3]) Consequently, this holds significant implications for the quality and reliability of end-user developed applications. This has resulted in many individual occasions where faults or "bugs" in end-user developed applications have resulted in significant, and costly, errors in a user's business (Barr & Foley[4]).

End-user computing continues to be a growing trend (Guimaraes[5]), and is one which is unlikely to diminish in the future as more and more people become "computer literate" in the workplace. A recent survey of the UK's top 2000 IT consumers showed 12% of systems are already developed by end-users and this is expected to rise to 18% by 1994 (Conducted by Applied Learning for James Martin[3]). The issue of the quality of end-user developed applications is, thus, also likely to increase in significance, as even more crucial end-user applications are developed and used. The computer professional has always been aware of the need for adopting good practice and methodology in systems development. The introduction of Quality Management Systems (QMS), conforming to recognised International Standards (ISO[6]) can be considered as a formalisation of this previous good professional practice. End-users must address the quality issue in the applications which they develop. They can no longer rely on inadequate, or non-existent, development and documentation procedures.

This paper considers the issue of quality in end-user applications development, and proposes a framework aimed at enabling end-users to

develop their own software within an overall quality environment. This will include the specification of a set of relevant quality guidelines for application to end-user developed applications.

GROWTH AND BENEFITS OF END-USER COMPUTING

End-user computing, ie the concept of the end-users themselves developing their own applications, is not a wholly new idea which has arisen with the advent of the personal computer. According to Panko[1], end-user computing was first promoted by IBM in the early 1970s. At that time, management feedback indicated that end-user applications were small and simple and that support and control for them should be directed towards individuals.

End-user computing has been defined by Hong-Mon-Yen[7] as "... giving the user free control and latitude over the computing process, including the development and use of Information Systems Applications". Remenenyi[8] defined end-user computing as "... the managerial and professional use of computer power and is in contrast to the clerical tasks that may happen to use the same types of computer hardware but have a different purpose". He also identified 5 classical end-user computing applications,

- Word Processing
- Figure Work (using Spreadsheets)
- Record Storage, Update and Retrieval databases
- Graphics (from Spreadsheets or Graphics Packages)
- Communications (Electronic Mail, Information Databases or Facsimile)

According to Guimaraes (Guimaraes[5]), the growing trend towards end-user computing has been caused primarily by the dramatic price and performance improvement in computer technology (epitomised by the PC), and the IS department's inability to satisfy user information requirements. Thus the initial set of application areas has grown (Panko[1]) and now includes,

- Personal Computing
- Decision Support Systems
- Executive Information Systems
- Non-Clerical Office Automation
- Work Station Host Data Communications
- Local Area Networking, including PC Networking

The long term importance of end-user computing was first realised when a statistical study in 1970 suggested that end-user computing was negligible. However by 1980 it had become 40% of total CPU cycles (Benjamin[9]). In a more recent NCC Members Survey, projections suggest the number of

mainframes will drop by 21% between 1991-1994 whilst the number of intelligent terminals is expected to rise by 89% during the same period (NCC[10])

The availability of the PC resource, however, cannot be the simple answer for the growth of end-user computing and the consequent shift of a significant portion of application development responsibility. Why, for example, can the end-users' needs not be satisfied by the existing IS department - where an organisation has one?

There have been some empirical studies, mostly in the U.S.A., of end-user computing and its impact on organisations who introduced it. For example, one advantage has been identified as being enhanced productivity of professional and white collar workers which overcame the shortage of DP professionals (Alain & Weiss[11]). Clearly, this indicates one significant reason for the growth in end-user computing, in that, the existing demand for IS services cannot be met by the existing numbers of personnel in IS departments. Would it help end-users if Information Systems (IS) departments employed more staff? It may be surprising, but it is doubtful if this would help. Another identified reason for its growth was that end-user computing has been identified as "... quicker and more responsive" than an IS department (Steele & Bottomley[12]). Accordingly, end-users found that the IS department required extensive lead time to make improvements and that some applications were never given the appropriate priority. Users viewed end-user computing as the best method of obtaining a system advancement or report that would otherwise be unavailable. Modifications and further improvements were (in their eyes) easily obtainable and implemented much faster.

This clearly indicates that, given the opportunity, end-users would generally rather develop their own applications (or at least certain types of application), rather than use the existing IS department. Other results from Guimaraes[5] found that users expressed greater confidence in reports generated by their own computing, and in general felt that they were better-off than when they were involved in traditional DP approaches.

Whether the traditional IS community like it or not, end-user computing is a phenomena which is unlikely to diminish. Indeed, quite the opposite is the case. The majority of users feel happier, and see more benefit, in adopting end-user computing as part of an IT strategy. It seems plain, therefore, that in the future there will be two separate forms of developed applications - those developed by IS professionals and those developed by end-users. None can be viewed as more important, or more correct, than the other. Both will have to be treated with equal priority and care.

QUALITY ISSUES IN END-USER COMPUTING

With the significant increase in end-user developed software and its related wider PC usage, a number of new, important issues have arisen. These issues can be grouped into 3 main categories. If an organisation wishes to implement a successful end-user environment, these 3 areas must be explored and addressed. The first area concerns the process of application development itself, the second concerns the overall strategic questions relating to end-user computing for the organisation as a whole, and the third deals with the worries end-users have themselves on matters such as training and security.

The end-user application development process

It can be suggested that the power of the personal computer lies in its potential to liberate users to be more productive and creative. Yet the PC has given rise to a whole new breed of user, a new category of proliferation of corporate waste and inefficiency, and in some cases, a lethal weapon. Of concern to management in some organisations is the number of end-users whose "business critical" application projects are being built and implemented with little or no consideration for elementary controls, no application of standard development methodology, no backup/recovery/security procedures, no validation or testing, no co-ordinated effort to link developments to specific business goals, and expensive duplication of effort and technology (Moad[13]). This new position raises the issue of control and its consequent effect on productivity. Unrestrained end-user computing can lead to inconsistent and incoherent systems which themselves inhibit rather that enhance user productivity. On the other hand, an over-restrictive regime can generate an atmosphere of too many rules, leading to limited computer use. Appropriate control, on the other hand, can enhance productivity. End-user computing must be encouraged and directed by pro-active management and yet controlled so that over-consuming budgets, inaccurate solutions to problems and negative internal relations (features associated with earlier DP systems) do not occur. The paradoxical issue facing many organisations keen to promote the notion of end-user computing centres on finding a balance between productivity and control.

Organisational and strategic issues in end-use computing

There is also a lack of a concrete corporate strategy for end-user computing generally and end-user developed applications in particular. Indeed, formalised implementation plans for end-user computing rarely exist. Management are preoccupied with reaping the benefits of end-user computing; they are relatively uninformed of the importance of a formal end-user computing implementation plan. In many cases this results in a repair of the problems created by not following an established methodology or a set of formalised procedures in the first place (Benson[14])

As a result end-users are not aware of the information which is available or how their requirements fit with those of other organisational units or functional areas. Additionally lack of prioritising in application development leads to scheduling conflicts, duplication of effort and unshareable data. With no overall plan, end-users develop their personal agendas for the development of applications, with their own criteria for data integrity and security. As a result, inappropriate solutions for organisational problems, redundant applications, inconsistent data and the development of critical systems with little concern for the needs of other stakeholders, are developed. The risks of end-user computing are accelerated by the fact that most end-user computing developed applications are not for personal use, but according to Rockart and Flannery[2], more than half of the systems surveyed concerned applications relevant to the operations of an entire department. Indeed 17% of the systems involved multiple departments and multiple functions.

End-user concerns

Apprehension in the area of end-user computing does not lie exclusively in the realms of senior management. In a study of a Washington based Home Loan Corporation, undertaken by Steele and Bottomley[12], many concerns about end-user computing were cited when employees were interviewed. When examined, some of these concerns are significant in nature and involve:-

Slow development of competent and appropriate solutions to business problems. Most users had little familiarity with the available end-user computing tools. Their choice was usually on a basis of familiarity rather than information requirements. Inexperience and weak selection methods generally resulted in an inefficient development process. Few users abided by structured development methods; testing and documentation were minimal; most of the systems were poorly backed up; data dictionaries and program libraries were practically unheard of. Documentation was indeed, an unknown word.

Ease of modification, although an advantage, also gives cause for concern. Organisations can become dangerously dependent on employees who have modified applications. As documentation for the application is generally inadequate, the danger is intensified. The use of backup disks usually begins after some accidental loss of a complete program, of a disk full of files (Benson[14]). Disks are commonly kept at random in a desk drawer or in a file box beside the computer. Backup disks are often kept in the same file box, so when this is stolen or lost, no other copy exists.

Time spent learning new systems or debugging programs seemed to outweigh the benefits of end-user computing. Debugging time and time spent solving

system problems could be reduced if users more clearly understood when to obtain assistance from the Information Systems Department(BIRD[15]).

Lack of security. Information for decision making must be of good quality, safe, and retrievable only by authorised individuals. Access to corporate extracts from production systems was inadequately controlled, and data of a confidential nature could easily be accessed.

One area which the Steele and Bottomley[12] study did not stress was the lack of testing involved in End-user developed applications. David Lewis, a specialist in end-user computing has "never seen anyone test a spreadsheet"(BRAY[3]). Indeed testing end-user applications using a prepared test plan, he believes, is almost unheard of. With no formal plan it is impossible to re-test an application after it has been altered. Frequently end-users fail to spot their own mistakes and if one person completes the whole application, there is in fact no-one to cross check the results. Often testing is seen as a boring and mundane task. Consequently with no specification to indicate it must be carried out, end-users frequently carry out a few obvious tests and hope no major blunder occurs. Employing a QMS for end-users would attempt to address some of these issues.

THE REQUIREMENT FOR SOFTWARE QUALITY MANAGEMENT IN END-USER COMPUTING

The case for quality management in the area of end-user developed applications is clearly established. If productivity is to mature in the organisation as a whole, control of this area must be established, unmistakable and easily enforced. If the foregoing discussion does not indicate the need for a suitable quality management system for end-user developed applications, then reference to BS5750 should illustrate the matter clearly. If an organisation has applied the standard to their IS developed applications, it cannot claim to satisfy the standard if their significant end-user community does not comply with this software quality management system. This section attempts to discuss the relationship between the accepted software quality standard and its application to end-user computing.

Software can be defined as "intellectual creation comprising the programs, procedures, rules and associated documentation pertaining to the operation of a data processing system" (ISO[6]). Clearly, end-users are, by defintion, developing software and as such their applications (software) should come under the locus of this standard. Organisations are finding they have not developed standards and policies which relate to the software design process in the end-user computing environment. If rigorous standards were to be

applied to this area, it may even prevent end-user systems being developed in the first place. Many end-users are relatively ignorant of the power of the new breed of computers and have limited skills in using them. Consequently they have no appreciation of the need for formal software design methodologies.

Failure to apply systematic design principles can have a profound impact on organisations and end-user productivity. For example, inherent conceptual design risks associated with "on the fly" design techniques could lead to inappropriate problem representation and use of unfit implementations. The risks associated with poor (or non-existent) documentation are all too familiar. Equally, failure to do even minimal testing, leaves the company at great risk and symbolises why a quality assurance policy must be applied. A generally accepted principle found in most organisations relates to maintenance of software. Maintenance of application systems is often expensive and time consuming due to short-sighted implementations of "quick and dirty" solutions

Quality in software development is increasingly becoming a fundamental issue for organisations regardless of size. Much of the IT world is quietly acknowledging that IT departments are applying too narrow a definition of quality (reference). They tend to measure systems solely in terms of the specified functionality - ie if they do the things they are supposed to do. However, in reality, a good system should satisfy a much wider range of criteria. If IT professionals are limiting their definition of quality, is it realistic to expect non professionals to produce "quality" systems when such a narrow criteria of quality standards exist? Indeed in the area of end-user computing, no recognised quality standards and procedures exist at all for developers to follow. In essence, users are give a "free hand" to produce software to whatever they believe to be "acceptable quality".

In contrast to this, standardisation of procedures continues to progress as an established part of the industrial and commercial world through the operation of British Standard 5750. The BS5750 standard is a Quality Management System (QMS) not a product standard. Nor is it a guarantee of product quality. A QMS is an administrative system which identifies, formalises and establishes standards for the regulation of processes found in any business. Quality Management is implemented through a Quality System, the latter being a collection of management and quality-related practices which are considered necessary for projects and derived tasks.

In applying this standard (BS5750) to software, and in particular end-user developed software, a key concept is that "a software development project should be organised according to a life cycle model". It is not prescriptive about the specific model to be used. However it is specific about the

milestones and deliverables of each stageof the project. The documentation relating to the deliverables form a comprehensive set of material which not only constitutes a record of the development activities but also serves as an essential tool in subsequent maintenance once the application is in use. The entire project must be monitored by appropriate management to ensure compliance with the quality system.

Clearly, whilst the use of a simplified life-cycle model is appropriate to end-user developed applications, a number of the milestones are not. For example parts of the Contract Reviews from the standard can be modified to produce a simpler proposal review at which the status of the proposed application would be determined. As with the ISO 9000-3 standard, where the specific details of the system procedures and its documentation are left to the judgment of each supplier, so too should the details of user-developed applications be left to the judgement of the individual organisation. All that can be suggested here is a minimum set of guidelines for a framework within which the details of a QMS for end-user developed applications can be constructed, and these form the basis of the next section.

PROPOSED FRAMEWORK FOR DEVELOPMENT OF A QUALITY SYSTEM FOR EUC DEVELOPED APPLICATIONS

There are no absolute rules for fitting end-user computing into an organisation. However general principles relating to the development of new applications by end-users can be produced. Here, a simple framework shall be outlined and explained. Subsequently a number of general issues relating to its implementation are discussed.

Proposed Framework for End-User Developed Applications

1. Proposal review
2. Requirement specification
3. Project schedule
4. Design and implementation
5. Testing
6. Application installation
7. Maintenance

Proposal Review

One of the first decisions to be made concerning a proposed end-user development is whether or not it should come under the control of the QMS. It is not suggested here that every foray with a PC must be done via the QMS. On the contrary, end-users will invariably develop a wide variety of application models, ranging from a simple spreadsheet to be used for the presentation of information, to a larger database of sensitive business data being used and updated by several departments. Guidelines must be developed by each organisation concerning the "nature and scope" of applications which must be developed through the QMS. Whilst, the precise details of these are organisation dependent, a simple categorisation of "critical" or "non-critical" application, could assist the proposed development. The definition of "critical" could be standardised and a "risk co-efficient" factor generated which would help the user and the organisation decide "who" should develop the application. However, as a general rule, if an application is processing "business critical" data, or is to be used by several users, either within a single department or across departments, its development must follow the procedures of the QMS. In order to resolve any disputes or to make these decisions, it is important that each department has a designated individual (a Quality Agent) who is given the overall responsibility for the operation of the QMS within his/her department.

Requirements Specification

End-users should present to this Quality Agent, a complete unambiguous set of functional requirements prior to proceeding with the development of those applications coming under the charge of the QMS. To ensure cohesive functionality, this may involve a dialogue with other potential users of the system. Consideration should be given to the inputs, outputs, processes and controls within the project being developed. The specification should be reviewed prior to approval being granted to proceed to the development stage.

Project Schedule

A simple timetable should be produced by the user and approved by the Quality Agent, which specifies tasks and stages of development. This schedule should include timescales and the allocation of individuals where the projects involves several end-users in its construction. The schedule should specify required inputs and outputs and the timing of progress reviews.

Design and Implementation

A suitable design methodology for use in end-user development must be specified by the organisation and form part of the QMS. The methodology need not be, indeed should not be, a rigorous one, in contrast to those used in large-scale professional developed systems. It should allow the presentation of a design in a top down fashion which could then be translated into suitable

code. Organisations should also develop simple coding standards for at least the main categories of application software tools they use, to assist in the readability and understanding of code. This would also assist in subsequent maintenance of the application.

Testing

Adequate test plans must be formulated which relate to the original specification and cover all aspects of its defined functionality (operational characteristics, ability to undergo change, and adaptability to new environments). Due regard should be given to specific aspects of testing such as the recording of test results, discovered problems and their effect on implementation.

Application Installation

At this stage the product should be ready for live running. Before this can be effected, some element of pilot testing in a "live" setting must be undertaken. In readiness for this, appropriate user documentation must be written and training provided to target users. A final review of the application will be required at the end of this stage before approval can be given for complete live running.

Maintenance

A simple procedure for the recording of identified problems and further enhancements should be developed. An individual responsible for resolving these for each application must be identified. All changes made during maintenance must be adequately documented.

General support issues

Responsibility for some aspects of the end-user environment (eg education/training, hardware standards) belong to the Information Centre (where one exists), while others rest with user management. However because of the nature of the individual working in this area, it must be accepted that a greater degree of support must be provided to this category in order to ensure good practice and ultimate success of the QMS. This support could take various forms but suggested elements would be:-

Training in basic software methodology. The typical end-user is a specialist in their particular professional field. A basic IT awareness may be assumed but even a superficial knowledge of the stages in software development cannot be presupposed. It is therefore essential, for training in areas such as basic design techniques and test-plan construction to be provided, either in-house or by external consultants. Such a training course would normally be of 3-4 days duration, be sufficiently comprehensive yet not superficial in treatment.

Departmental support systems. With the possibility of several users developing/using these programs, it will be necessary to provide them with a departmentally based support system to assist them with the collation and organisation of security copies, update procedures, document control and maintenance and housekeeping procedures. This may involve the creation of a departmental based library facility under the management of the departmental quality agent.

Organisational support activities. With the adoption of QMS at departmental level there requires to be a degree of co-ordination and support provided at organisational level. Management commitment to this activity therefore is a crucial factor in the formulation and standardisation of this operation. They should establish and resource an End-User Quality Group. This group would include each departmental agent as representatives. The members will formulate policy, provide guidelines, monitor and review the system at appropriate intervals to ensure its continued suitability and effectiveness. In addition management should set up a centralised specialised support group consisting of suitably qualified and experienced IT personnel. This group should provide expert assistance to individual end-users where necessary. These "experts" would provide a range of advice on such diverse subjects as, selecting suitable software tools for developing proposed applications, or acting as troubleshooter in solving particular programming problems end-users may encounter. If organisations have an existing IS facility, this role can be undertaken by them. If not, senior management must recruit appropriate personnel to provide this level of support on an organisation wide basis.

SUMMARY

This paper has asserted that end-user computing continues to be a growing phenomenon unlikely to diminish over the forseeable future. Consequently, more software will be developed by the users themselves. This development activity must be suitable managed. Accordingly businesses should consider formally developing a simple but well managed quality management system for end-user developed applications. Responsibility of these application projects should be assigned to an appropriate named person within the organisation. This responsibility should include reviewing end-user proposals to determine if they are considered "critical" and should accordingly come under the charge of the Quality Management System. If they do, a development plan should be initiated and managed. In all cases, project applications should conform to the organisations "guidelines for end-user developed applications".

REFERENCES

1. Panko, Raymond P , End User Computing: Management , Applications and Technology; Wiley (1988)

2. Rockart, John F., and Lauren S Flannery. "The Management of End-User Computing", Communications of the ACM, Vol 26 No 10, October 1983 Pg 776-84.

3. Bray, Paul, "The Software Time Bomb" Which Computer, September 1992.

4. Barr, Stephen, & Foley, Richard, "End Users First", CA Magazine, March 1993.

5. Guimaraes, Tor, "The Benefits and Problems of User Computing" , Journal of Information Systems Management 1984, (fall), pg 3 - 9.

6. Department of Trade and Industry, "TickIT, Guide to Software Quality Management System Construction and Certification using EN29001", 28 February 1992, Issue 2.

7. Hong-Mon-Yen, Ching-Cha Hsieh & Ming-Te-Tu, "Implementing End User Computing in Smaller Firms" , Industrial Management and Data Systems 90, 6.

8. Remenyi, Dan, Management Accounting (UK), Vol:68, Iss 11, pg 31, Dec 1990

9. Benjamin, Robert I., "Information Technology in the 1990: A Long Range Planning Scenario," MIS Quarterly, June 1982, pg.11-31.

10 NCC's Ninth Survey of Class B Membership, December 1991.

11. Alain, Maryam & Weiss, Ira " Managing the Risks Associated With End-User Computing". Journal of Management Information Systems, Vol 2 No 3. pg5-20 Winter 1985-86.

12. Steele, Jean & Bottomley, Dick, "Channeling the End-User Computing Initiative : A Case Study.

13. Moad, Jeff "The Second Wave", Datamation Feb 1989 pg 14-15.

14. Benson, David, "A Field Study of End-User Computing : Findings and Issues" MIS Quarterly, Dec 1983 ,pg 35 - 45.

15. Bird, J. Management Today, Feb 1992, "Managing Information Tehnology, Micro Myopia".