

A template to assess user centredness in software quality management

A.M. Smith, L. Dunckley

*Department of Computing and Information Systems,
University of Luton, Park Square, Luton, LU1 3JU, UK*

Abstract

User-centredness is now generally perceived as an important ingredient in the successful management of software projects. There is, however, some lack of clarity within the software development community as to what specifically constitutes a user-centred design process and how organisations can seek to increase their user-centredness. The authors have previously identified three categories of *structures*, *processes* and *scope* through which user-centredness can be evaluated. These categories have previously been used to analyse and present the results of a major study of user organisations. From the work of this study the authors have developed a template to assess user centredness in software quality management. In this paper the authors describes the design and use of such a template. When applied by software development managers, the template provides a mechanism for objective assessment of user-centredness. The authors presents the results of trialling the template in a number of user organisations. In conclusion it is suggested that the template offers software development managers a rapid way to assess their design methods and identify weaknesses. By adopting strategies for remedial action overall user-centredness and thereby software quality can be enhanced.

1 Introduction

A number of IT projects have recently come to national attention supporting the assertion that IT systems would still appear to be failing because of a lack of effective user engagement. Historically, systems developers have often considered user involvement as a necessary evil. Despite this user-centredness is now perceived as an important ingredient in the successful management of

338 Software Quality Management

software projects. In a previous study (Smith, 1993) a major survey of UK commercial user organisations was undertaken in order to identify to what extent, and in which ways, genuine User Centred Design (UCD) principles have been integrated into mainstream commercial IT systems design. This study represents the most significant body of evidence available within the United Kingdom to date as to the extent of, and problems associated with UCD. From the survey it was possible to identify which particular aspects of UCD are being adopted at a faster rate than others and to ascertain a number of factors which influence the degree of user-centredness in both UK organisations and IT projects.

Through the survey it emerged that software development managers have a commitment to user involvement which goes further than mere functional requirement elicitation. User-centred principles are, however, not being adopted uniformly across the IT industry. The great majority of designers do not adopt sufficient tools and techniques to enable active user involvement and therefore do not apply a genuine user-centred design approach. There is certainly a disparity in the speed and depth of the adoption of the different aspects of UCD. It emerged that approximately one fifth of organisations / projects demonstrate a genuine practical user-centred approach.

The survey highlighted the lack of a shared understanding within the software development community about what constitutes UCD and how individuals might address any weakness in their approach. In response to this the authors have developed a self-assessment template which enables individual software developers to evaluate their user-centredness.

The template has three specific aims:

- (i) to raise awareness within the design community about what constitutes user centred design
- (ii) to enable individuals and organisations to identify their strengths and weaknesses in respect of UCD
- (ii) to provide a sign-posting facility which will direct software development managers to the many tools and techniques which are currently available to integrate human factors within the development process.

2 User Centred Design in the UK

There is now a very large number of design methodologies available to the software developer and user-centredness of such methods is well documented

(Holman, 1988, Robbins 1989, 1991). Methodologies currently in use which are designed to address user and organisational issues include ETHICS (Mumford, 1983) and Soft Systems Methodology (Checkland, 1981,1988). Other structured and post-structured methods in use within the UK include SSADM (CCTA, 1990), JSD (Jackson, 1983) and Object Oriented Design (Booch, 1991). A range of tools designed to address human factor issues are also available to top-up the design method at various stages in the life cycle. These include methods for Task Analysis (Fountain, 1985, Shepherd, 1989, Eason and Harker, 1980) and participative methods for user communication such as the use of story boarding and scenario techniques. Approaches to prototyping are also important. In addition tool-set approaches such as HUFIT PAS (Galer et. al. 1989) are available.

There is a very limited body of evidence available in the literature relating to UK take up of UCD techniques. Two sources are, however, of note. Firstly, Hornby and Clegg (1992) describe a study of the processes of participation during the design and implementation of a new computer-based information systems at a large UK bank. The essential finding was that although software development managers claimed that they were committed to the goal of effective user-participation, the end user's view was rather different. In their Productivity Enhancement Programme (PEP) the Butler Cox Foundation (now CSC Index) have investigated the rather more limited user involvement in systems development. Green (1992) reveals that 98 per cent of organisations 'consider the involvement of users to be an important aspect of their development practice' and that over 70 per cent claim to 'use methods that support user involvement' but that under 20 per cent support 'active' involvement.

3 Towards a framework for evaluating UCD

It is suggested that approaches to user-centred design can be addressed under the following broad categories: *structures*, *processes*, and *scope*. It is these categories which have been used, within both the template and the earlier survey, to analyse approaches to UCD:

Structures to support the design process include project management, design team structures, and mechanisms for user involvement and communication between the design team and the user community.

Processes for design focus around the design methodology and include issues relating to requirement elicitation, communication and implementation, including approaches to prototyping.

The ***scope*** of the design process relates to how far the analysis reflects a

340 Software Quality Management

socio-technical, as opposed to just a technical, solution. More specifically an IT project with only a narrow scope will focus on technical and functional requirements whereas one with a wider scope will address issues such as the allocation of functions between man and machine, the design of work structures and individual jobs and the ways of enhancing job satisfaction within the organisation.

4 The state of play of UCD in the UK

In the earlier survey (Smith, 1993) a range of medium to large UK organisations participated by completing a postal questionnaire and taking part in follow-up interviews. The questionnaire contained questions grouped under the following sections: design methods, project management, user requirements and organisational implications. Each of the questions was mapped to one of the UCD categories (structures, processes and scope). Questions relating to each category occurred in a number of sections within the questionnaire. A Principal Components Analysis of the results of the survey was performed which identified that the main factors within the survey environment were user focus, iterativeness, techno-centredness and communication structures.

The survey showed that most progress towards user-centredness is being made within the area of structures to support the overall process. Structures were found to be more extensive at the organisational and formal representative level, rather than at the detailed local design level. The problems associated with further developing the use of user oriented structures relate to the difficulty in recruiting appropriate, interested and skilled personnel who have the time available for project secondment.

Whilst there is considerable evidence that traditional design methods still dominate the design process, the survey indicated that a wide variety of other methods, providing a range of user engagement, are in use although there is no evidence of the take up of specific participative methodologies. Whilst the shift to structured methods has come about mainly for reasons of efficiency and effectiveness within the design process it has also led to a significant increase in user involvement. Within the identification of user requirements emphasis is given to high level issues (e.g. Business Mission and Strategy) and less so to low level ones (e.g. Task Execution). Whilst there is evidence of methods for task analysis and for the use scenarios or dramas / storyboarding the majority of software developers use no formal method for eliciting or specifying requirements, relying exclusively on verbal interviews and natural language recording mechanisms. Prototyping was found to be the most well supported UCD topic. Whilst the main reason for prototyping was to ensure system functionality, it was interesting to note that prototyping was also used for

interface design and for the non-technical issues of work structures and processes.

Organisational issues is the area where least progress is being made towards a user centred approach. Whilst over two thirds cite partnership between users and designers to be important only one fifth are specifically able to integrate social and technical issues. There is evidence that design staff are aware of the need to widen the scope of design but feel that organisational priorities and senior management policies negate against them making progress.

Clearly not all of the UK IT industry is approaching UCD in the same way or at the same pace. Firstly, as discussed above, there is a disparity in the speed and depth of the adoption of the different categories within UCD. Whilst structures are increasingly common, and processes are developing, organisational issues are largely ignored. The second consideration influencing the uniformity of introduction relate to factors inherent in the specific company and project. A large number of influences are present but in particular it would appear that companies providing a customer service are more user-centred. Cross-organisation projects also lead to more user-centredness. It seems likely that there is a relation between the interactiveness of the project and the degree of user involvement.

Figure 1
An Example Question from Template Version 1

Please indicate the **one** statement which is most applicable to the way software projects are normally developed in your organisation.

1. Project design and implementation is normally led :
 - a) by the user department
 - b) through a partnership between the user department and central MIS/DP
 - c) by central MIS/DP
 - d) by external consultants

5 Development of the UCD Templates

Overall the survey indicated that only approximately one fifth of organisations / projects demonstrate a practical user-centred approach. It is in order to address this weakness within systems development that the authors have developed the self-assessment templates. Two versions of the template are currently available.

342 Software Quality Management

The first focuses on providing a quantitative indication of UCD whilst the second provides a more qualitative analysis derived from the principal components analysis of the survey results.

5.1 Template Version 1

The first version has been developed directly from the survey instrument and leads to an objective analysis of user-centredness under each of the three categories within UCD. Rapid self-assessment has been achieved by enhancing the structure inherent in the earlier survey instrument. For each of twenty questions, the user is required to select one answer from a list of four. An example question is shown in Figure 1. The answer selected for each question generates a score in the range 1 to 4. Each question maps to one of the categories within UCD. The user is directed to group the individual question scores under each of the categories and is thereby able to generate a score for each category and an overall score for user-centredness. Percentile values from the earlier survey sample have been calculated so that the software developer is able to determine whether he/she is in a low, middle or high score group. The template provides a commentary on what the scores indicate and how remedial action may be taken. The facility for comparison against population percentile values provides a strong assessment method which will be further refined as implementation of the template is extended.

5.2 Template Version 2

A potentially more powerful version of the template has been developed. Version 2 adopts an approach similar to that developed by Belbin (1981) for a

Figure 2
Software Developer Types

Type	Characterising Statement
A	I try to involve as many users as possible
E	I try to avoid contact with users
B	I adopt structures to support the user
F	I do not provide mechanisms which would integrate users
C	I use a range of user centred processes
G	I mainly use processes which address technical issues
D	My approach is socio-technical
H	My approach is primarily technical

self perception inventory for management teams. This version of the template allows for an enhanced qualitative comparison between the approaches to each of the categories within UCD. Here the structure category has been divided into two categories representing the range of users involved and the structures implemented to support them. These four categories together with their opposites represent eight different types of software developer which are described in Figure 2. When completing this template the software developer is presented with 5 sections and is required to distribute a total of ten marks between eight responses to a statement. For each statement, such as shown in Figure 3, an response is provided which might be typical of each of the eight Software Developer Types (A - H). By recording scores in an analysis sheet the user is able to calculate an index score for each of the developer types and plot these on a User Centred Profile. A completed example is presented in Figure 5. Version 2 does not currently enable easy quantitative comparison with other software developers although this facility will be available after further implementation.

Figure 3
An Example Section from Template Version 2

For each section below distribute 10 points among the sentences you think best describe your views. The points can be distributed among several sentences; in extreme cases they might be spread amongst all the sentences.

Section 1

From my experience of IT project design and implementation:

- (a) I consider the establishment of Local Design Groups with full user involvement to be important in the management of successful projects
- (b) I have found that success can only be achieved by establishing a mechanism whereby all potential user groups can be involved in the development process
- (c) I have found that successful project design and implementation is best led by central DP/IS departments
- (d) I have found that it is not necessary practice to consider the design of jobs for employees
- (e) I believe that the best approach to prototyping is to not use it at all
- (f) Where user requirements are concerned I know that formal methods such as GOMS and Hierarchical Task Analysis can be used successfully
- (g) I consider that when creating a design team structure it is sufficient to ensure that users are available for requirement elicitation
- (h) I consider that when identifying user requirements for a project the future impact on organisational structures is highly important



6 Trialling and Further Development

Both versions of the template have been trailed within a limited number of UK organisations. One aim of trialling was to elicit the views of software developers concerning ease of use and clarity in both the methods for UCD assessment and the ways of identifying remedial action. Trialling has also enabled an initial comparison between analyses produced by each of the templates. From the results of trialling it emerged that the templates were easy to complete and provide mechanisms for clear identification of user-centredness. Further work is currently in progress which will refine the facility which provides information concerning remedial action.

The authors would not expect identical results to emerge from the two templates; Version 1 was developed from the initial survey whilst Version 2 was designed independently. However if the templates are to be of direct use, particularly in identifying remedial action, they would need to provide consistent results. For each organisation the results from the two templates were found to be generally compatible. Organisation A, to take an example, was placed in the high score group for all three categories of user-centredness as indicated by Version 1 (see Figure 4).

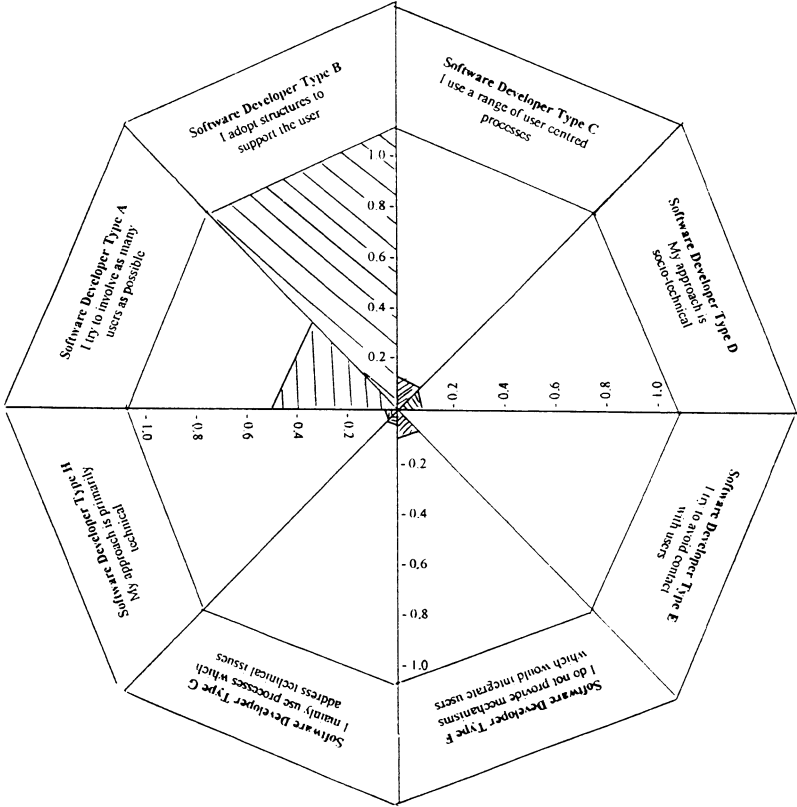
Figure 4
Results from Organisation A

Template Version 1		
Category	Score	Percentile Range
Structures	17	95 -100
Processes	24	60-65
Scope	15	70-75
Template Version 2		
Software Developer Type	Index Score	
A	0.51	Average Index Score Types A - D = 0.42
B	1.0	
C	0.11	
D	0.07	
E	0	Average Index Score Types E - H = 0.03
F	0.07	
G	0.03	
H	0.03	



A comparable result was achieved in Version 2 with the average index score for Software Developer Types A - D (positive user-centredness) and E - H (negative user-centredness) of 0.42 and 0.03 respectively. Version 1 indicated that structures were the most strongly adopted category (in the 95-100 percentile range from initial survey sample) which was supported in Version 2 with Type B (adopt structures) generating the highest index score. In general Version 2 was found to magnify differences in approach; for example Version 1 rated processes and scope only slightly less highly than structures whereas Version 2 indicates a greater difference. It would, of course, not be the intention for organisations to complete both templates at the same time. Version 2 is designed to highlight a qualitative comparison between the Developer Types. The authors are currently distributing both versions of the

Figure 5
User Centred Profile



346 Software Quality Management

template to a wide range of organisations for further testing and development.

7 Conclusions

Impact analyses of IT systems indicate a relatively low success rate. Studies in North America suggests that only 20% of systems implemented achieve something like their intended benefits, 40% fail and the remaining 40% make only a marginal impact on the organisation. More limited, but more recent, studies in the UK are not totally at odds with these findings. User acceptance is a crucial factor in increasing success rates. Effective user centred design is a significant factor in maximising user acceptance. The templates have been shown to be an efficient and effective tool to raise awareness of UCD, provide a mechanism for self-assessment and indicate areas of potential remedial action. Through wider distribution, implementation and development it is suggested that they can make a contribution to the spread of UCD techniques and could therefore contribute to the overall enhancement of software quality. The templates are available from the authors at the University of Luton.

8 References

- Belbin, R.** 1981, Management Teams. (Heinemann, London)
- Booch, G.** 1991, Object Oriented Design with Applications. (Benjamin Cummings, Menlo Park CA)
- CCTA.** 1990, SSADM Version 4 Reference Manual. (NCC Blackwell)
- Checkland, P. B.** 1988, Soft Systems Approaches and their Integration into the Development Process. In Contemporary Ergonomics, Proceedings of the Ergonomics Society's 1988 Annual Conference. (Taylor and Francis, London)
- Checkland, P.** 1981, Systems Thinking, Systems Practice. (J Wiley)
- Eason, K. and Harker, S.** 1980. An open systems approach to task analysis. Internal Report. HUSAT Research Centre. Loughborough University of Technology
- Fountain, A. J.** 1985, Modelling User Behaviour with Formal Grammar. In People and Computers: Designing the Interface. Proceedings of the British Computer Society Human Computer Interaction Specialist Group. (Cambridge University Press)
- Galer, M, Taylor, B et al** 1989, The HUFIT Toolset. Human Factors in IT. In Contemporary Ergonomics, Proceedings of the Ergonomics Society's 1989 Annual Conference (Taylor and Francis, London)
- Green, P.** 1992, Involving Users in Systems Development, Butler Cox Product Enhancement Programme Paper 19 August 1991. (CSC Index, London)
- Holman, R.** 1988, Integrating Human Factors into the Systematic Development Process. The State of the Art and the Future. In Contemporary Ergonomics, Proc. of the Ergonomics Society's 1988 Annual Conference (Taylor and Francis, London)

- Hornby, P. et al** 1992, a, User Participation in Context: A Case Study in a UK Bank. In Behaviour and IT, 1992, Vol. 11, No 5. (Taylor and Francis, London)
- Mumford, E.** 1983, Participative Systems Design: Practice and Theory. In Journal of Occupational Behaviour. Vol. 4. 1983.
- Robbins, I.** 1989, The Contribution of Structured Systems Design. In Participation in Systems Development: UNICOM Applied IT Reports. Ed. Ken Knight (Kogan Page)
- Robson, J,** et al 1991, Systems Analysis and Design Methodologies: Are these methods addressing Human Issues. In Contemporary Ergonomics, Proceedings of the Ergonomics Society's 1991 Annual Conference. (Taylor and Francis, London)
- Shepherd, A.** 1989, Analysis and Training in Information Technology Tasks. In Task analysis for Human-Computer Interaction. Ed Dan Diaper (Ellis Horwood)
- Smith, A.** 1993, A Survey of User Centred Design. M.Sc. Thesis, Interactive Computing Systems Design, Loughborough University of Technology