A SYSTEMIC MODEL FOR PARTICIPATORY CHANGE IN INFORMATION SYSTEMS MANAGEMENT.

Andy Hyde

Clinical R&D
Nycomed Amersham PLC
Nycoveien 1-2
PO Box 4220 Torshov
N-0401 Oslo
Norway

1. INTRODUCTION

Computers, whether we like it or not, are a growing part of our daily existence. As with any technological advance throughout history it brings with it great change. Many people fear change of any kind but where does this fear come from? In IS I believe it has something to do with a feeling of a lack of control. Computer users become more aware of the possibilities for each new generation leaving education and joining the workforce. The days when a system developer knew more than the user are fast disappearing. A new systems development methodology is required that puts the developer and user on the same level.

Participatory methods of development do exist but have been more a theory than a practice. I present here a development methodology based on the principle that people who have the possibility to change their own future and experience that change as positive when compared with their mental model of the world will willingly and enthusiastically participate in that change. For this, the process of change is more important than the outcome because if the user participation has been effective, whatever the outcome, it will be correct.

The methodology presented in this paper has a systems epistemology through the feeding back of positive experience and new knowledge into the user's mental model so that these can affect future decisions. I suggest the use of contextual evolutionary prototyping as a vehicle for providing the required feedback whilst developing the IS solution.

2. THREE GENERATIONS OF DEVELOPMENT METHODOLOGIES

Somebody once said “We do not fear change but we fear *being* changed”. These words succinctly describe the situation that the computer revolution has created in modern society. The change that computers have brought has happened at a pace which for many has been too fast. Individuals as well as society as a whole needs time to absorb change, to understand it and become a part of it otherwise it can develop into fear.

We do not fear change if we are a part of that change. The rate of development in Information Technology and Information Systems has perhaps left more people behind and created more fear than many previous changes in society because of its penetration into so much of our daily existence. The change has been driven largely by ‘experts’. The domain of computer science has been a closed domain to outsiders, but that is changing.

Benathy (1997) has discussed these changes in the context of social systems but the same can be seen in Information Systems based on computer technology. Benathy identifies three generations: Developing *for* the users, developing *with* the users and developing *as* a user.

The first generation of developing *for* the users in the IS context was the ‘expert’ deciding what the user needed, developing and delivering a completed system. This, in the early days of the computer where there were few educated in the new science was perhaps understandable.

The second generation of developing *with* the users started as, and to a large extent is still, more an impression than a reality. A short consultation may be the most interaction that takes place between a developer and the user before the system is developed and delivered. Methodologies for development based on Participatory Design (PD) have made a genuine attempt to involve the user. These methodologies however have lacked a teachable theoretical background and have therefore not become part of computer education.

The third generation is developing *as* a user. In the social systems discussed by Benathy we are all part of the system being designed. With Information Systems the technological part of the system is external, we interact with it but are not a part of it. The parallel though is that the developer of the system should become part of the domain in which the user exists and into which the new system will be placed. From this position the developer can understand how the user perceives the system from his or her own view of the world, not as is most often the case, from the developer’s external view of the world. One way to achieve this is as Shackel (1990) suggests, that the developer becomes a user. This approach is similar to Ethnography, more commonly associated with sciences such as anthropology, but which is seen as having more relevance to the study of Human-Computer Interaction (Preece *et al*, 1994)

If the developer is developing the system from within the user’s domain and not from outside, he or she is in a much better position to see the users as individuals and account for this in the change process.

3. DEVELOPING IN THE NEW GENERATION

Many IS projects still fail and much research has been done to identify the reasons for failure. Traditional reasons are given as time and cost overrun and failure of the system to perform according to specification, however, these can be seen as emergent from two other reasons for failure; a failure to manage the complexity (Kokol, 1997; White, 1997) and a failure to manage expectations (Bennetts and Wood-Harper, 1997). Failure to manage complexity will lead to time and cost overrun and failure to manage expectations will lead to the system not performing according to specification. The latter because the specification is built on an expectation of what the system will be able to do.

Methods of development in the early second generation are still used. Little or no contact with users is sought by the developer before detailed design and implementation begins. In traditional development methodologies a complete description of the required system, the system specification, should be created before any design work is done. Beeby *et al*. (1997) believe that this is not possible. The requirements will be generated over time as the development progresses. From this view it is clear that a running dialog between user and developer is required throughout the process.

A conceptual definition and understanding of what is being developed is required. An Information System can be viewed as many things to many people. Kammersgaard (1990) describes these views as perspectives. Two of these are especially pertinent to this discussion. The system perspective and tools perspective. The system perspective is that most often seen by the technologist. The new system being developed is a technological achievement. The system is comprised of computers, networks, software and interfaces. The tools perspective, however, looks at the development as an extension of the tools available in the user's domain, the tools necessary for employees to carry out their daily tasks. System developers need to realise this and see their role as the development of a tool for the users that will require constant refinement through iterative design and testing. Developers need to see the knowledge and skill that individual users posses as essential to that development.

Users can contribute constructively, contrary to the beliefs of many IS developers. Drucker (1998) succinctly captures this realisation in the context of worker involvement in management in saying that he and others were surprised to find that workers were neither "Dumb oxen nor immature nor maladjusted" as others had maintained earlier.

4. A NEW METHODOLOGY

The theoretical underpinning of the methodology is to provide constant feedback to enable change at an absorbable pace. For users to be able to absorb the change it is necessary to understand how the user perceives the world of technology and technological change. Each individual has a different perception based on their mental model of the world. This mental model is based on the individual's motivation, values, visions, knowledge, experience, and often assumptions and guesses. Oppenheim (1992) suggests that action is determined by attitudes which are only abstractions - "though they are real enough to the person who holds them". This point is central. Each user's attitude to another may appear unfounded but still the system developer must address the issue. It may not be possible to build a system that encompasses every individual's needs but each must have been addressed to the satisfaction of user who perceived it otherwise the user's mental model will be negatively reinforced. The mental model is used to filter information before action is taken and the results of the actions are also used to shape the mental model. Bad experiences can therefore be a block to action but good experiences can be a catalyst.

The methodology suggested in this paper has five elements. The elements each contribute to experiential learning by the constant use of feedback during the development allowing the users to fully understand the implications of the change and the way the change is being implemented, to influence the change and absorb the change.

The elements alone and in some combinations are not new, but as with Senge's five disciplines (Senge, 1993) it is the synergy achieved from the combination of all five that makes the methodology what it is.

The elements are:

1. Ethnographic Requirements Analysis
2. Participative development
3. Contextual development
4. Evolutionary prototyping
5. Systems Thinking

4.1 Ethnographic Requirements Analysis

In order for the system developer to obtain a full understanding of the domain into which the new system will be placed and an understanding of the users it is necessary for the developer to be within the user domain for more time than is traditionally the case. The developer needs to spend time looking at the environment, the conflicting tasks, the organisation and the individual users. The belief that the user will behave the same as the developer or even that one user will behave like another is described by Landauer (1990) as “naïve intuition fallacy”. Davenport (1994) observes that technocrats are constantly caught off guard by the “irrational” behaviour of end users. Only by close and constant interaction can these misunderstandings be corrected.

4.2 Participative Development

PD is traditionally short for Participative Design but Chin *et al*. (1997) suggest there is a need to involve the users much earlier in the process. This view is shared by Mumford (1997) who has developed the QUICKethics methodology to involve users in the requirements analysis phase. Carmel *et al*. (1993) and Blomberg and Henderson (1990) found that effective participation is not just inviting users to a design meeting, the users need to be involved and need to perceive their input as having meaning and effect. The term Participatory Development is therefore used to emphasise this requirement as distinct from the detailed design phase where other participatory methodologies tend to focus.

PD has no fixed definition but Blomberg and Henderson (1990) have identified three tenets that guide practitioners: the goal is to improve the quality of working life, the orientation is collaborative and the process is iterative. These three tenets underpin the methodology described in this paper.

The methodology is inescapably participative because of the contextual evolutionary prototyping and the way that the requirements are developed iteratively from an ethnological interaction.

4.3 Contextual Development

The tradition in the design of computer systems for human use through Human-Computer Interaction techniques has been to use laboratory testing to design a usable system. However, designing in a laboratory does not provide for feedback based on real world conditions. Whiteside *et al*. (1988) highlights several failings of the laboratory approach based on environmental differences and task design and Landauer (1990) questions it on psychological grounds. Checkland and Scholes (1990) in describing action research point out that this requires involvement in the problem situation.

If one is to understand how the new development will interact with other aspects of the user’s human activity systems, development must be done in the context of those systems. Systemic interactions of the technology cannot effectively be separated from other parts of the system.

4.4 Evolutionary Prototyping

Evolutionary prototyping will enable the iterative development of requirements and the iterative design and testing of the new tool whilst allowing the users to absorb the change.

Established methods of Participative Design (PD) described in Clement and Van den Besselaar (1993) and Blomberg and Henderson (1990) use passive methods of design such as paper based scenarios to create a design. However, Beeby *et al*. (1997) emphasise that the best way for users to understand the change is by experiencing it first hand.

The methodology's epistemology is based on the need for experiential learning. Everybody in the change process needs to learn from each other and from the experience in order to change their mental model of the world.

4.5 Systems Thinking

No system can be developed separate from its environment and organisation. Soft Systems Methodology (SSM) (Checkland and Scholes, 1990) clearly emphasises this in the SSM that evolved from experience of its use. The seven stage model was replaced by the model with a cultural stream of analysis and a logic based stream of analysis. The logical based stream of analysis in the outlined methodology of this paper is the contextual evolutionary prototyping. Mental based analysis is replaced with action research based experimentation. The implications of change resulting from the implementation of Information Systems is complex, just the type of complexity SSM sets out to address, perhaps too complex to be fully understood before it is experienced.

5. CONCLUSIONS

The methodology is developed out of the need to involve the users in the process of change and in the belief that the development of Information Systems which are the tools of our daily work can only be done effectively with the synergies resulting from a co-operation between users and developers.

The developer must develop a better understanding of the user's domain and the user must learn, at an absorbable pace, the capabilities of the technology used to build Information Systems. The effects of implementing technology based Information Systems are increasingly complex, especially when a systemic view of the world is held. It is therefore suggested that a new methodology with a systems epistemology is required.

The methodology places the developer in the user's domain to teach and to learn. The development project is incremental in order to avoid the generation of fear. The development must be in the context of the final implementation so that it will function in the real world when completed. Although, because the world keeps changing, the development is never complete until it is replaced by a successor.

The percentage of new technology projects that fail is alarmingly high, even after so many years of experience. Analyses of the reasons for these failures consistently point to problems of co-operation or communication between developer and user. If the user is alienated by the change process they are more likely to resist it or even sabotage it. The methodology suggested involves the user in a way where they can see that they have a real possibility to affect the change and therefore their future. By using positive reinforcement to emphasise this and replace bad experience with good and ignorance with knowledge it is hoped that users will want to become involved in the change process and will actively and constructively contribute.

REFERENCES

Beeby, R.B., Gammack, J.G., and Crowe, M.K., 1997, “Constructing End-User Design Environments: implementing client-led design”, in *Proceedings of the 5th conference of the United Kingdom Systems Society*, Stowell *et al*. eds., Plenum Press, N.Y., USA. pp. 537-541

Benathy, B., 1997, “Designing Social Systems in a Changing World: a journey to create our future”, *Systemist*, Vol.19, No.3, pp.187-216

Bennetts, P.D.C., and Wood-Harper, A.T., 1997, “Soft Systems Methodology: a metaphor for the process of data analysis”, in *Proceedings of the 5th conference of the United Kingdom Systems Society*, Stowell *et al*. eds., Plenum Press, N.Y., USA., pp.543-547

Blomberg, J.L., and Henderson, A., 1990, “Reflections on Participatory Design: lessons from the trillium experience” in *CHI ’90 Proceedings*, The Association for Computing Machinery, Inc., N.Y., USA, pp. 353-360

Carmel, E., Whitaker, R.D., and George, J.F., 1993, ‘PD and Joint Application Design: a Transatlantic Comparison’, *Communications of the ACM*, Vol.36, No.4, pp.40-47

Checkland, P., and Scholes, J., 1990, *Soft Systems Methodology in Action*, Wiley, Chichester, UK

Chin, G., Rosson, M.B., and Carroll, J.M., 1997, “Participatory Analysis: shared development of requirements from scenarios’, in *Proceedings of CHI ‘97 Electronic Publications*, The Association for Computing Machinery, Inc., N.Y., USA.

Clement, A., and Van den Besselaar, P., 1993, “A Retrospective Look at PD Projects”, *Communications of the ACM*, Vol. 36, No 4, pp 29-37

Davenport, T.H., 1994, “Saving IT’s Soul: human-centered information management’. *Harvard Business Review*. 1994 (March-April). pp. 119-131

Drucker, P.F., 1998, *On The Profession of Management*, Harvard Business School publishing, Boston, MA., USA.

Kammersgaard, J., 1990, “Four Different Perspectives on Human-Computer Interaction” in *Human-Computer Interaction*, Preece, J. and Keller, L. eds., Prentice Hall, Hemel Hempstead, England. pp.42-63

Kokol, P., 1997, “Reasoning about software system design with SSM”, in *Proceedings of the 5th conference of the United Kingdom Systems Society*, Stowell *et al*. eds., Plenum Press, N.Y., USA. pp.579-582

Landauer, T.K., 1990, “Relations Between Cognitive Psychology and Computer System Design”, in, *Human-Computer Interaction*, Preece, J. and Keller, L. eds., Prentice Hall, Hemel Hempstead, England. pp.141-160

Mumford, E., 1997, “Requirements Analysis for Information Systems: The QUICKethics approach”, in *Proceedings of the 5th conference of the United Kingdom Systems Society*, Stowell *et al*. eds., Plenum Press, N.Y., USA. pp.15-20

Oppenheim, A.N., 1992, *Questionnaire Design, Interviewing and Attitude Measurement*, Pinter, London, England

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., and Carey, T., (1994), *Human-Computer Interaction*, Addison-Wesley Publishing Co., Wokingham, England

Senge, P., 1993, *The Fifth Discipline*, Century Business, London, UK.

Shackel, B., 1990, “Human Factors and Usability”, in *Human-Computer Interaction*, Preece, J. and Keller, L. eds., Prentice Hall, Hemel Hempstead, UK.

White, D., 1997, “Risk Management and project failure” in *Proceedings of the 5th conference of the United Kingdom Systems Society*, Stowell *et al*. eds., Plenum Press, N.Y., USA. pp.525-529

Whiteside, J., Bennett, J, and Holtzblatt, K., 1988, “Usability Engineering: our experience and evolution”, in *Handbook of Human-Computer Interaction*, Helander, M. ed., North Holland, Amsterdam, Netherlands.