A BEHAVIOURAL ANALYSIS OF MODELS OF THE INFORMATION SYSTEMS DEVELOPMENT PROCESS

Susan Gasson,
Warwick Business School,
University of Warwick,
Coventry CV4 7AL, U.K.

1. INTRODUCTION

There is a strong argument that the use of structured methodologies to support information system development (ISD) leads to fragmented, highly specialised, low-discretion (i.e. deskilled) jobs for system users (Corbett et. al., 1991; Markus and Bjorn-Andersen, 1987). While deskilling is a work strategy which some managers may wish to pursue, many do not; structured methodologies are therefore inappropriate for many development projects. This paper is intended to stimulate debate on process models to support alternative methodologies; it is presented in the context of current research and also on the basis of the author's experience as a practitioner in the field of information system design.

![Diagram of the Waterfall Model of System Development](Source: based on Friedman & Cornford, 1989, page 290)

Corbett et. al. (1991) argue that the use of structured ISD methodologies leads to a cognitive approach centred upon scientific reductionism, as work tasks are successively decomposed until the developer can associate a computing algorithm with the sub-task. This approach causes extreme task fragmentation, as there is little consideration of user task identity or the user's mental model of the task performed. Successive decomposition is
employed because structured development methodologies are based upon the "waterfall" process model (Boehm, 1988) illustrated in figure 1, where each stage of the process represents a level of problem decomposition. The output of each stage is validated, then feeds into the next stage as input, with feedback permitted only between contiguous stages. The waterfall model has been criticised for its over-reliance on documentation as a measure of progress (Boehm, 1988) and for a lack of integration of the "system" of user knowledge into the information system design (Land & Kennedy-McGregor, 1987). Successive decomposition has been rejected by other areas of creative design, such as architecture, as being unrepresentative of "real-world" design processes (Lawson, 1980). Yet the waterfall model still forms the basis for most information systems development methods in current use in the UK (Eason, 1982; Moynihan & O'Connor, 1991).

Alternative development approaches, such as evolutionary development (Eason, 1982), the ETHICS approach (Mumford, 1983), Multiview (Avison & Wood-Harper, 1990) and Change Analysis (Goldkuhl & Rostlinger, 1993) have been developed to combat the limitations described above. However, these approaches concentrate upon development methods, which are largely contingent upon organisational context and upon reflective (usually academic) practitioners; they do not provide alternative models of the processes engaged in by IS professionals with limited time and resources. Lyytinen (1987) argues that both traditional and new ISD methodologies lack or have limited theoretical foundations: while theory is borrowed from other fields such as Artificial Intelligence or Semantics, such theory is inappropriate to the "hazards of social change" (Lyytinen, 1987, p 5). It is critical therefore, to derive models of the development process which can be operationalised into methodologies to support a more human-centred development approach.

2. ALTERNATIVE MODELS OF THE DESIGN PROCESS

Alternative design approaches, which have been proposed to overcome one or more of these deficiencies, may be seen to fall into three broad categories:
A) EVOLUTIONARY DEVELOPMENT

Prototyping is considered here as a tool for development, rather than as a methodology, which prescribes a sequence of process methods. A typical evolutionary development model is shown in figure 2: this consists of a series of short-timescale project lifecycle iterations which may involve the creation of system prototypes, or may consist of staged deliveries of system versions with evolving functionality.

![Figure 2: A Typical Evolutionary Development Process Model](Source: Elaborated from Smith & Mayes, 1992)

One of the major problems in any IS development project is that the further into the development lifecycle the project is, the more the design becomes "frozen", leaving little freedom for changes to the specification as users become more familiar with the new system technology from their contact with the development project process (Eason, 1982). Apart from the lack of underlying theory discussed in the previous section, a particular criticism of evolutionary process models is that they may lead to an early freezing of unnecessary design constraints because of limitations in the scope of early prototypes or system definitions, leading to the loss of a system overview (Sol, 1984; Boehm, 1988).

While this process may allow the user to give feedback on system usability and function, the scope and definition of that system (and the human activity system which it supports) are
largely left to IS professionals. The system requirements definition is identified as the most critical stage of the process (Galliers, 1987) in terms of the impact of constraints imposed upon the design at this point, yet this stage is seen as the least defined of most prototyping and evolutionary methodologies (Boehm, 1988; Floyd, 1984), leading to a tendency in IS professionals to start with a prototype, rather than a thorough system requirements analysis.

B) PROCESS CONTROL

This area is concerned with process management issues at a macro level; Boehm's (1988) spiral model of software development, given in figure 3, is seen as a prominent attempt to manage uncertainty and risk in ISD project management (Curtis et al., 1988). In this model, the radial dimension represents the cumulative cost of development to date, the angular dimension represents the progress made in completing each cycle of the spiral.

![Figure 3: The Spiral Model Of Software Development](Source: Boehm, 1988)

An underlying concept of this model is that each cycle involves a progression that addresses the same sequence of steps, for each portion of the product and for each of its levels of elaboration. However, this model does not address the interconnectedness of the
steps in system definition and design, nor does it address behavioural issues, such as the
cognitive and social processes of IS development.

C) SOCIO-TECHNICAL AND HUMAN-ACTIVITY MODELLING APPROACHES

Methodologies in this class rely more on the expertise and understanding of the
practitioner than on a clearly defined process-model. Examples of methodologies which use
this type of approach are Soft Systems Methodology (Checkland & Scholes, 1990), which
stimulates much debate on what are valid process steps for an SSM analysis (e.g. Mingers,
1992), the ETHICS method (Mumford, 1983) and the Multiview approach (Avison & Wood-
Harper, 1990), significant frameworks which address the issues of ISD requirements analysis,
but do not address the behavioural processes involved in their implementation.

3. BEHAVIOURAL RESEARCH INTO INFORMATION SYSTEMS DEVELOPMENT

3.1 LEVELS OF PROCESS CONTEXT

Curtis et. al. (1988) comment that the effects of tools and methods can be seen to be
relatively small compared to the impact of behavioural (human and organisational) factors on
software productivity. An effective model of ISD processes must therefore support
behavioural factors as well as technical ones. Curtis et. al. (1988) propose the layered
behavioural model shown in figure 4.

![Figure 4: The Layered Behavioural Model Of Software Development
(Source: Curtis et. al., 1988)](image-url)
In this model, they propose three behavioural levels of analysis: the individual (cognitive and motivation), the group and the organisation. These levels will be used as the basis for a behavioural model of ISD; this section discusses process model issues at each level and their implications for ISD methodologies.

3.2 THE INDIVIDUAL LEVEL

The underlying assumption of the waterfall process model is that the cognitive processes of design follow Simon's model (Newell & Simon, 1972) which separates design into three consecutive stages: intelligence (problem analysis), design (outlining and evaluation of alternative solutions) and choice between alternative solutions. In this "top-down" model, a problem is identified and successively decomposed into sub-problems until a sufficient level of decomposition is reached to permit a solution synthesis.

Friedman & Cornford (1989) suggest that most development follows a model of progress which cycles between non-contiguous levels of problem decomposition; Lawson (1980) hypothesised that, for people with previous design experience, analysis is more closely integrated with synthesis than in the top-down model. Turner (1987) sees design as a bottom-up process, where requirements and solutions migrate together towards convergence. In synthesis, the design process can be seen to be a combination of top-down (problem decomposition) and bottom-up (solution synthesis) activities (figure 5).

![Figure 5: A Convergent Model Of The Design Process](Source: figure derived from Turner's (1987) analysis of Malhotra et. al. (1980))

Malhotra et. al. (1980) conclude that information system design actually takes place on many levels of problem decomposition at the same time; the design process is constrained by the existence of requirements implicitly perceived by a designer; requirements which are not
surfaced by the design process until an implicit requirement conflicts with an explicit requirement. This has implications for structured methodologies which record only requirements at the current, "official" level of problem decomposition.

Jeffries et. al. (1981) argue that the type and suitability of a solution proposed by a designer rely upon an individual's design schema: an understanding of the methods of design and of possible solutions for certain types of problems derived from the designer's individual background, education and experience. The empirical studies of Curtis et. al. (1988) appear to support this argument; they conclude that the success of ISD projects depends upon an "expert designer" - a senior team member with a high degree of previous development experience in the current application domain, whose function is to educate other team members in ways of solving problems in this application domain.

The issue of individual motivation is also of importance here: studies have shown (Markus, 1984; McNurlin & Sprague, 1989) that IS professionals have a relatively low social-orientation, whereas system users are likely to have a much higher need for social interaction, and that IS professionals have a relatively high personal growth need leading to an excessive need to "play with" new technology. IS professionals are therefore much more likely to optimise the technical aspects of the system at the expense of the social, since they feel most comfortable with technical issues.

3.3 GROUP DESIGN PROCESSES

Ciborra (1987) defines the main problem of IS design as arising from design methods' focus on individual decision-making rather than on "human interaction and exchange" within organisations. A valid process model must take account of social interactions: the elements of communications and of learning which take place during the ISD process.

Curtis et. al. (1988) observe that the lack of support for effective communications provided by development methodologies leads to a subversion of "official" procedures, such as design reviews, to facilitate improved communications between IS professionals in different development teams, but that communication can easily be thwarted as the methods used do not support it.
Walz et. al. (1987) comment: "While individual design tasks are largely cognitive in nature, the process of designing software when no individual possesses all the knowledge and/or skills required, also includes a dimension of interactions."

Rosenbrock (1981) suggests that engineers and designers learn a *normative* approach to problem-solving through the group processes of education and negotiation provided by the project design team. It is possible that this may evolve into a shared version of the design schema discussed above: the process of validating design decisions against other teams' interpretation of this group design schema may be central to development team communications.

The importance of the learning process in design is raised by the empirical studies of Curtis et. al. (1988) in their analysis of the "expert designer" role discussed above, and by those of Turner (1987), who observed student groups solving practical design problems. The dominant strategy observed by Turner for problem-solving was for the group to attempt to derive a set of rules which described the situation, even when they were told that this was counter-productive prior to the exercise. In any model of group design processes, both individual and shared learning are *central* to the process; this must be recognised by allocating project time for this activity. Development team members acquire application domain knowledge from users (Curtis et. al., 1988) and users acquire technical and system knowledge from development team members (Eason, 1982). If the ISD methodology does not facilitate this transfer of knowledge, both developers and users are reliant upon different models of the system, its users, its methods of use and its purposes; these may be untrustworthy and may have a high degree of incompatibility.

### 3.4 ORGANISATIONAL PROCESSES

This level is concerned with progress management of ISD projects and with interfaces between the development team and other groups not normally involved in the development process such as marketing personnel or the firm's customers.

Curtis et. al. (1988) comment that one of the most significant challenges to the success of large development projects is that of coordinating communications from different customer
sources to provide a consistent understanding of requirements. As for the group communications discussed above, clear communication channels to actors outside the development project must be considered essential to shared understanding of the system and to learning processes.

In the area of progress management, any model of ISD processes must recognise the iterative nature of design and development. While Boehm's (1988) spiral model provides progress control-points at each cycle, the sub-phases of this model show the cycles to be less interrelated than the empirical research, discussed above, would suggest. In contrast, design is perceived to be an evolutionary activity (Eason, 1982). The solution evolves with increasing awareness of the organisational situation, the product-market, the application domain and pressure from influences external to the development team (such as changing user requirements or the commercial needs of the firm's customers); it evolves at a multiplicity of decompositional levels simultaneously.

4. A BEHAVIOURAL MODEL OF ISD PROCESSES

A behavioural model for information systems development processes is presented in figure 5.
This model attempts to relate the behavioural processes of ISD to a staged (waterfall model) methodology, to show those points at which this type of methodology fails to support the process and also so that the implications for progress management become apparent.

The main implications of the new model are:

- the formal process paths defined by structured development methodologies only comprise a small part of the actual process paths during ISD. Assumptions and decisions are unrecorded when they occur on process paths which are not covered for formal methodologies
- a large part of this process is implicit, therefore the surfacing of implicit requirements needs to be supported by development methodologies, to make the process more effective.

There are cognitive references to design schemas which embody assumptions and experience at both an individual and group level; it may be possible to be recognise and challenge these schemas during information system development.
- requirements exist at all levels of problem decomposition at the same time and process iterations may not be undertaken to resolve successive levels of decomposition, but to clarify areas of conflict between explicit and implicit requirements; these iterations need to be documented in order to record the basis for design which may need to be communicated (for example to developers taking over later stages of the project) or tested to validate the system

- as iterations do not reflect levels of decomposition, measuring progress on the basis of levels of decomposition (as with the waterfall model) is meaningless: progress measures need to be developed which evaluate the extent of target system synthesis

- requirements do not automatically fall out of the process of design; they are arise from shared understandings both explicitly negotiated with and implicitly learned from other participants in the design process. These understandings provide a definition of system (in the wider sense of organisational system) meanings and values: they should be recorded and questioned as part of the system validation process and supported by the establishment of explicit and clearly defined communications channels.

The implications for future research are that a behavioural investigation of requirements definition and negotiation is central to an understanding of the development process and that any operational model of development processes must function at multiple levels of behaviour.

The above analysis indicates the importance of incorporating human-centred values into the development process for information systems. To include users in the process without the questioning of development values is not sufficient. To quote Gill (1991):

"Given the long-standing and deep tradition of splitting [the structural separation of IS professionals from users], being "human-centred" in attitude and assertion is not enough. Methodologies are needed for operationalising it."

REFERENCES
Checkland, P. & Scholes, J.(1990), Soft Systems Methodology In Action, Wiley, Chichester


