ICIS 1998 Proceedings

International Conference on Information Systems (ICIS)

December 1998

An Empirically-Grounded Framework for the Information Systems Development Process

Brian Fitzgerald
University College Cork

AN EMPIRICALLY-GROUNDED FRAMEWORK FOR THE INFORMATION SYSTEMS DEVELOPMENT PROCESS

Brian Fitzgerald
Executive Systems Research Centre
University College Cork
Ireland

Abstract

It is generally taken as axiomatic that systems development methodologies play a useful role in guiding the systems development process, and that their increased adoption would improve the process and product of systems development. This paper summarizes the arguments and pressures which support the use of methodologies. However, the problems associated with the use of methodologies have not perhaps received as much attention in previous research. A number of factors are identified in the paper which question the value of methodologies. These dichotomous arguments—in favor of and against the use of methodologies—bring about a dilemma for systems developers as to whether to adopt a formalized development methodology or not. This research study was therefore concerned with investigating the nature of systems development and the use and role of systems development methodologies in actual practice. The study adopted a comprehensive multi-phased and pluralist research strategy. First, the researcher's commercial experience as a systems developer, coupled with a detailed examination of the literature, and a pilot study involving personal interviews with experienced developers in five organizations were drawn upon to validate and refine the research objective. The next stage of the research involved a postal survey of developers in 776 organizations. Quantitative statistical techniques were used to analyze the survey responses. Following this, the final phase of the research involved a field study comprising personal interviews with 16 experienced practitioners in eight organizations. The field interview data was analyzed in a qualitative manner so as to elucidate and elaborate on the survey findings. Based on the research findings, the paper presents an empirically-grounded framework which describes the development process as it pertains to modern organizations. The components of the framework are discussed in detail and relevant implications are identified.

Keywords: Information systems, information systems development, IS development methodologies, IS development approaches, pluralist research.

1. INTRODUCTION

The literature has traditionally viewed methodologies as axiomatically appropriate to improving both the process and product of systems development with the use of methodologies being typically seen as "positive and well advised" (Jenkins, Naumann and Wetherbe 1984). While there are a number of significant arguments in favor of methodologies, there are also a number of arguments and pressures that question the use of methodologies. These are discussed in detail in Fitzgerald (1996), and are summarized briefly in section 2 to serve as a point of departure for the study. Following this, section 3 outlines the research objective and overall research approach adopted for the study. Section 4 presents the framework for the IS development process which is grounded in the research findings. Each component of the framework is discussed in some detail in section 5. Finally in section 6, the implications of the framework are identified and conclusions drawn.

Table 1. Arguments and Pressures That Support the Use of Methodologies

Arguments That Support the Use of Methodologies
• Systems development is a very complex process. Methodologies may provide a reductionist subdivision of this process into plausible and coherent steps (Olerup 1991).
• By rendering the development task more visible and transparent, methodologies may facilitate project management and control of the development process, thus reducing risk and uncertainty (Ahituv, Hadass and Neumann 1984; Avison and Fitzgerald 1995; Floyd 1987; Friedman 1989).
• They may provide a purposeful framework for the application of techniques and resources at appropriate times during the development process (Ahituv, Hadass and Neumann 1984; Bantleman and Jones 1984).
• There is an economic rationale, in that methodologies may allow skill specialization and division of labor (e.g., analysis, design, coding and testing) which can receive different remuneration rates (Avison and Fitzgerald 1995; Friedman 1989).
• An epistemological rationale can be identified as methodologies may provide a structural framework for the acquisition of knowledge. Thus, any learning from past development experiences can be systematized and stored for future reference (Baskerville, Travis and Truex 1992; Stage 1991; Stolterman 1994).
• Standardization of the development process is afforded. This facilitates interchangeability among developers. Also, it can lead to increased productivity and quality, as resource requirements can be predicted and made available as and when necessary (Avison and Fitzgerald 1995; Downs, Clare and Coe 1992).
Pressures for Increased Use of Methodologies
• Desirability of ISO-certification in many organizations is fueling the adoption of methodologies as a means of achieving such certification
• Governments, which are significantly involved in systems development, have in some cases declared that certain methodologies be used for development. For example, SSADM (UK, Ireland, Malta, Hong Kong, Israel); Dafne (Italy); Merise (France); and NIAM (Netherlands). This creates a significant pressure in practice for the adoption of these methodologies (Downs, Clare and Coe 1992; Holloway 1989).

2. ARGUMENTS AND PRESSURES IN FAVOR OF AND AGAINST THE USE OF METHODOLOGIES

A number of significant arguments can be made in favor of the use of methodologies. In addition, there are a number of significant pressures, complementing these factors, which support the increased use of methodologies. These are summarized in Table 1.

It is not surprising given the arguments and pressures identified in Table 1, that there is quite a prevalent view that the use of methodologies is positive and well-advised, and the implicit expectation is that methodology usage should increase over time. Notwithstanding this, it has been acknowledged that practitioners have been somewhat slow in adopting development methodologies (Aaen 1986; Avison and Fitzgerald 1995; Page-Jones 1991; Ward 1991). Systems development methodologies are attractive and have an intuitive appeal, but a systems development methodology is merely a framework for organizing the system development process. Indeed, a number of significant arguments may also be advanced which serve to question the value of development methodologies. Also, just as there are a number of pressures that support increased formalism in the development process, there are a number of significant pressures for radical changes in the traditional approach to development. These are summarized in Table 2.

Table 2. Arguments and Pressures Against the Use of Methodologies

Arguments Against the Use of Methodologies
• Estimates suggest that more than a thousand brand-named methodologies exist (Jayaratna 1994). However, it is open to question as to whether there are that many meaningfully different ways to develop information systems. Thus, there are artificially contrived differences between some methodologies—the many variations of the structured approach, for example—while there are fundamental differences between others; for example, "soft" methodologies from the Scandinavian tradition differ greatly from "hard" functionalist methodologies such as the structured approach (Avison and Fitzgerald 1995; Lyytinen 1987).
• Generalization without adequate conceptual and empirical foundation as many methodologies have been proffered on the basis of limited use, without the twin customary safeguards of (1) expert independent review to determine their appropriateness and (2) their successful application on a non-trivial real-world case (Lyytinen 1987; Ward 1991)
• Systems development is not actually an orderly rational process but most methodologies view it as such, with a major emphasis on technical rationality at the expense of softer, social aspects (Baskerville, Travis and Truex 1992; Floyd 1987; Wastell and Newman 1993).
• There may be some means-ends inversion, as developers proceed in slavish and blind adherence to methodologies, while losing sight of the fact that development of an actual system is the real objective (DeGrace and Stahl 1990).
• Often there is an assumption that methodologies are universally applicable across all development situations—the one size fits all presumption (Chikofsy 1989). However, researchers have pointed out that due consideration needs to be given to the contingencies of each development situation (Avison, Fitzgerald and Wood-Harper 1988; Curtis, Krasner and Iscoe 1988).
• There is an inadequate recognition of developer-embodied factors in methodologies, as they do not cater for factors critical to successful development, such as individual creativity and intuition, or learning over time (Boehm 1981; Brooks 1987; Vitalari and Dickson 1983).
Pressures Against the Use of Methodologies
• At a broad level, the changing nature of the business environment means that short-term needs now dominate. Thus, traditional life cycle approaches to development, which result in the eventual delivery of systems after several years are no longer appropriate (Baskerville, Travis and Truex 1992; Brown 1985).
• There is also some evidence of an altered profile of the systems development environment. In-house development of systems may no longer be the norm. Rather, much development now consists of what could be labeled "configuration development," i.e., the integration and customization of package software to incorporate local practices (Bansler and Havn 1994).
• Also, there are moves toward rapid application development (RAD), with the replacement of large-scale monolithic approaches with alternatives based on the "frequent tangible results" philosophy of delivering some functioning components of the system in shorter time scales (Folkes and Stubenvoll 1992; Hough 1993).

3. RESEARCH OBJECTIVE AND RESEARCH APPROACH

3.1 Research Objective

These dichotomous arguments—in favor of and against the use of methodologies—bring about a dilemma for systems developers as to whether to adopt a formalized development methodology or not. However, the factors identified in Table 2 indicate that the assumption that methodologies are axiomatically an appropriate solution to development problems is at least open to question. The objective of this research was therefore to investigate the actual nature of systems development in real organizational situations, with a view to clarifying the use and role of methodologies in systems development practice.

This research objective required a comprehensive research approach. Certain questions, such as those related to the level of methodology adoption and the profile of the development environment, could be investigated quantitatively through a large-scale postal survey. However, other questions, such as those to do with the nature of usage and the roles played by methodologies, required a more qualitative, in-depth interview approach. Thus, the research contained a large element of methodological pluralism, albeit within an overall interpretivist research paradigm. Also, given the comprehensive nature of the research, the findings in respect to various aspects of the research have been published already. This paper attempts to provide a summary of the overall research, and, rather than attempting to incorporate the relevant findings here in what would be an overly-truncated fashion, references are provided to the relevant publications where the methodological issues to do with each research phase are discussed in more depth.

3.2 Research Approach

A model of the overall research process involved in this study is presented in Figure 1. The model depicts the manner in which the literature and the researcher's practical experience of systems development were drawn on to define the initial research objective, and continued to inform the research as it unfolded. A pluralist multi-phase research strategy was adopted. This strategy enabled the researcher to build up a large amount of research data to derive an empirically-grounded framework for the development process. The first research phase was a pilot study that involved personal interviews with experienced developers in five organizations in Ireland (Fitzgerald 1994). This served to validate and refine the research objective, and also helped in the construction of a postal survey questionnaire.

The second phase involved a postal survey of a total of 776 named individuals in different organizations who were likely to be directly involved or responsible for systems development. The findings of the survey have been extensively reported elsewhere (Fitzgerald 1998). However, a postal survey cannot probe deeply into all research issues. Consequently, the final research phase involved in-depth personal interviews with 16 experienced developers in eight organizations. A primarily qualitative research strategy was adopted for the interview research phase, with the researcher leveraging the fact that he was a "cultural insider" (Bodker and Pedersen 1991), having

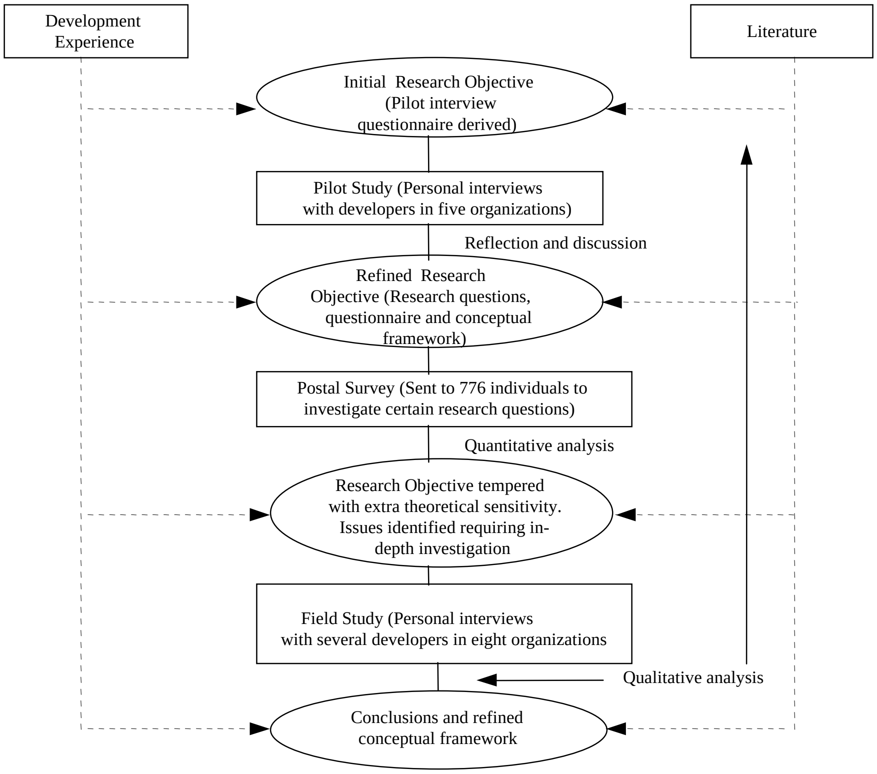
Development Experience
Literature
Initial Research Objective (Pilot interview questionnaire derived)
Pilot Study (Personal interviews with developers in five organizations)
Reflection and discussion
Refined Research Objective (Research questions, questionnaire and conceptual framework)
Postal Survey (Sent to 776 individuals to investigate certain research questions)
Quantitative analysis
Research Objective tempered with extra theoretical sensitivity. Issues identified requiring in-depth investigation
Field Study (Personal interviews with several developers in eight organizations
Qualitative analysis
Conclusions and refined conceptual framework

Figure 1. Summary of the Research Approach

been a professional systems developer for a number of years. Again, the findings of the interview research phase are discussed in detail in Fitzgerald (1997).

4. INTERPRETATION OF RESEARCH FINDINGS

The necessity for a conceptual research framework to guide the research process has been captured well by Checkland (1994, p. 7):

> If you are to take part in a change process and learn from it in a research sense...it is essential to declare in advance the intellectual framework in terms of which what counts as learning will be defined....That is a condition for moving beyond the anecdotal.

A framework was initially derived, based on the findings from the pilot study, the literature review, and the researcher's practical experience as a developer (Fitzgerald 1995). To quote Miles and Huberman (1984), this represented the "researcher's map of the territory being investigated." However, Miles and Huberman also make the point that qualitative research allows for modification of the framework as the researcher conducts the research. Therefore, the framework has been refined (see Figure 2) to reflect the findings from the survey and field interview research phases.

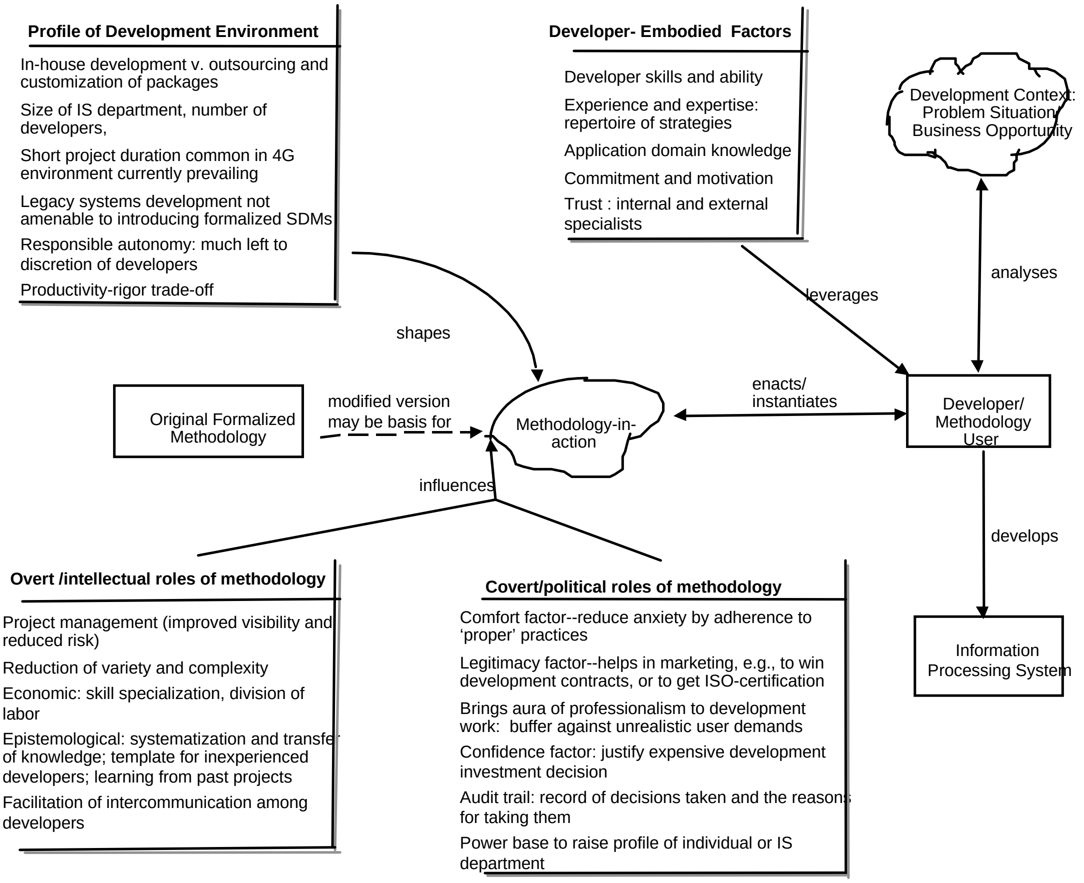
Profile of Development Environment
In-house development v. outsourcing and customization of packages
Size of IS department, number of developers,
Short project duration common in 4G environment currently prevailing
Legacy systems development not amenable to introducing formalized SDMs
Responsible autonomy: much left to discretion of developers
Productivity-rigor trade-off
Developer- Embodied Factors
Developer skills and ability
Experience and expertise: repertoire of strategies
Application domain knowledge
Commitment and motivation
Trust : internal and external specialists
Development Context: Problem Situation/ Business Opportunity
analyses
leverages
shapes
Original Formalized Methodology
modified version may be basis for
Methodology-in-action
enacts/ instantiates
Developer/ Methodology User
influences
develops
Overt /intellectual roles of methodology
Project management (improved visibility and reduced risk)
Reduction of variety and complexity
Economic: skill specialization, division of labor
Epistemological: systematization and transfer of knowledge; template for inexperienced developers; learning from past projects
Facilitation of intercommunication among developers
Covert/political roles of methodology
Comfort factor--reduce anxiety by adherence to 'proper' practices
Legitimacy factor--helps in marketing, e.g., to win development contracts, or to get ISO-certification
Brings aura of professionalism to development work: buffer against unrealistic user demands
Confidence factor: justify expensive development investment decision
Audit trail: record of decisions taken and the reasons for taking them
Power base to raise profile of individual or IS department
Information Processing System

Figure 2. Framework for the IS Development Process

When one proposes any model or framework, it is useful to recall George Box's extremely apt observation that while "all models are wrong, some are useful" (Box 1979, p. 202). The framework in Figure 2, by necessity, simplifies certain aspects of systems development. However, it is useful as a means of representing certain significant aspects of the phenomenon of interest, and helping communicate the findings of the research. As already mentioned, even though the research study was a pluralist one incorporating both quantitative and qualitative methods, the overall research approach was primarily an interpretive one. Thus, the framework should not be viewed as a predictive deterministic mechanism in a positivist sense. Rather, it serves to facilitate an understanding of the phenomenon in practice and its insights will primarily provide a basis for continued empirical investigations (cf., Lee 1991, Orlikowski 1991).

5. DISCUSSION OF THE FRAMEWORK COMPONENTS

5.1 Original Formalized Methodology vs. Methodology-in-Action

The framework makes the distinction between the original methodology as proposed by its creator and the *methodology-in-action* as interpreted by the developer. This is indicated by the dashed line in Figure 2, which reflects the view that while some kind of methodology will be used, a formalized methodology *may* be the basis, but is not essential. This distinction between an original formalized methodology and a methodology-in-action has parallels with the distinction drawn by Argyris and Schön (1974) between an "espoused theory" and a "theory-in-use." Thus, it is suggested that methodologies are never applied exactly as originally intended. Different developers will not interpret and apply the same methodology in the same way; nor will the same developer apply the same methodology in the same way in different development situations. Therefore, on any development project, the methodology-in-action is uniquely *enacted* or *instantiated* by the developer.

The view of a methodology-in-action, uniquely enacted for each development project, was very evident in the study, with support for this interpretation coming from a number of factors. First, it emerged in the survey that companies that stated they were using the same formalized methodologies did not converge in their actual development practices. Also, the survey found that only 6% of respondents stated that they were rigorously following a formalized commercial methodology. In addition to this, in the interviews, it was found that none of the organizations that were using methodologies were following them rigorously, all the more significant given the fact that some of these organizations had specifically tuned their methodologies to the specific contingencies of their development situation. For example, in one software house studied, great emphasis was placed on issues such as configuration management, testing, and even telephone support, whereas in a government department, where the development approach was based on he same commercial methodology as that of the software house, major emphasis was placed on issues to do with project inititiation and outsourcing contract management. Also, of interest is the fact that it was suggested in several organizations that developers departed from the methodology in a conscious and deliberate fashion, rather than an arbitrary one. For example, one major bank supplemented its methodology with more sophisticated techniques in the area of cost-benefit analysis, as this was not seen as being adequately covered by its methodology.

5.2 Development Context: Business Opportunity and Problem Situation

The framework acknowledges the complex and dynamic nature of the development context, in terms of both problem situation and business opportunity. Researchers have tended to focus on the problem situation in the past (Checkland 1981). However, during the field interviews, an interviewee expressed the view that if the development context is viewed as a problem situation, development tends to be seen very much as a reactive issue. He acknowledged that such a category of development existed, but stressed that a more proactive category, whereby business opportunities were being identified for which systems development would be necessary, was also becoming common, especially if IS departments were to become more closely involved in the planning and strategy formulation process. A similar view was expressed by other interviewees who felt that IS departments needed to win back the confidence of senior management by being more proactive and business-focused, rather than remaining outside the business core and only consulted after business strategy had been determined in many cases. This issue is discussed again later when the political roles that methodologies may play are discussed.

5.3 Developer/Methodology User and Developer-Embodied Factors

The developer/methodology user is accorded a central role in the framework, thus reflecting the fact that it is people, not methodologies, who develop systems, and that the latter are merely frameworks which must be leavened by the wisdom of human developers if they are to be effective. In relation to this, the importance in the development process of developer-embodied factors is explicitly acknowledged, as these are leveraged by the developer during development. There are several strands of research in the literature that confirm the importance of developer factors. For example, huge variances in the skills and capabilities of different developers and a consequent impact on development productivity have been reported (Boehm 1981; Brooks 1987; Peters and Tripp 1977). Also, researchers have identified the importance of learning over time as developers increase their level of expertise and develop a repertoire of strategies to apply in different development situations (Vitalari and Dickson 1983). The significant contribution made by developer knowledge of the application problem domain has also been acknowledged (Davis and Olson 1985).

Developer factors were found to be rated as very important during interviews. There was a definite trend whereby developers tended to have their own specialist niche areas. Their skills and aptitudes in these areas were known to IS management and project leaders, and these were factored in when allocating development work. Another developer-embodied factor that was identified during the research was that of developer motivation or commitment. A project manager with a vast amount of development experience suggested that the commitment of individual developers to the task was the most significant factor in ensuring that systems were successfully completed. Other interviewees confirmed the importance of developer commitment and motivation, and it was generally felt that these arose from a number of sources. For example, the experience of working on projects with new technology was cited as a significant motivator, as was the opportunity to work more closely with top management.

Another factor that emerged as being relevant during the interviews was that of *trust* in individual developers. This manifested itself internally in organizations where, given the intimate knowledge of developer aptitudes, it was often the case that trusted developers were assigned responsibility for particularly critical development tasks and projects. However, trust was a factor in external relationships also, with some organizations allocating vital development roles to external specialists whose contribution was an essential component of development in these organizations.

5.4 Profile of Development Environment

The "profile of the development environment" factor emerged as significant from analysis of the survey findings, where it was found that methodologies tended not to be used when the levels of in-house development were low and the levels of outsourcing and customization of packages were high. Also, the survey found that methodologies were significantly more likely to be used in large organizations (more than 1,000 employees) with large IS departments (more than 20 personnel), and when the number of developers on a project was high (more than five), and the project duration was long (more than nine months). This is in keeping with the project management role that methodologies play, and would suggest that they facilitate intercommunication among developers and help control long projects.

These findings were further confirmed in the field interview phase of the research study. Long projects with many developers in large organizations with formal working practices were more likely to be carried out under the auspices of a formalized development methodology. Typically, methodologies were seen as something that could ensure development was carried out more rigorously, but the general view was that productivity would suffer if methodologies were introduced, and so many organizations were not willing to introduce them. There appeared to be a fairly common view that systems could be developed to a "good enough" standard that was adequate for the users and the purpose of the system. Thus, in these organizations, the payback from the use of methodologies was not perceived being as worth the effort.

In a similar vein, several interviewees also identified a general tendency toward shorter duration of development projects in their current development environment. Large projects were subdivided and systems were delivered incrementally. Part of the rationale for this was that the business could have changed totally within two years. Therefore, projects of such long duration were completely impractical.

Some interviewees expressed the view that formalized methodologies were more suited to third generation development environments, in that they were felt to be monolithic and cumbersome, and thus unsuited to the type of fourth generation environment currently prevailing in their organizations. Also, outsourcing and subcontracting of development were quite common in the organizations studied, with several organizations reporting a low opinion of the quality of in-house development. Thus, it appears that many organizations have lost both confidence and competence in relation to in-house development, and are quite reluctant to undertake major development projects solely in-house.

A lack of formality and a fluidity in hierarchical positions in IS departments was evident in some organizations, where different developers, regardless of position or title, assumed the role of project leader or, indeed, had a fair degree of autonomy in relation to development. This is interesting in that Friedman (1989) has traced the history of the management of development as having passed through a number of phases. The first of these, in the 1940s and 1950s, he characterizes as "loose responsible autonomy" in so far as developers, who were typically scientists, were developing systems for themselves, and so they were not answerable to traditional control mechanisms. Friedman argues that systems development has since been characterized by a move toward tighter management control of the development process. However, in this research study, a degree of responsible autonomy on the part of developers was evident. This autonomy in relation to their working environment may also have contributed significantly to the motivation of developers, discussed in an earlier section.

5.6 Roles of Methodology: Overt Intellectual vs. Covert Political

Finally, the framework in Figure 2 explicitly identifies two conflicting categories of roles that methodologies can play in the development process. There are a set of overt intellectual roles that form part of the conceptual basis and rationale behind the use of methodologies. These have been well researched in the literature (see Table 1). These roles were further confirmed empirically in the study. However, the study also identified a set of covert, political roles. These have not been the subject of much research in the literature. These roles are discussed in turn next.

5.6.1 Overt Intellectual Roles

First, methodologies play an important role in making the development process more amenable to project management and control by affording more transparency into the development process. Thus, there are milestones and deliverables which offer opportunities to review progress, thereby minimizing risk. This potential role of methodology was supported by interviewees in all the organizations and was also a factor that received a high level of agreement from respondents in the survey.

A further intellectual role is the economic rationale. This relates to the skill specialization and division of labor, which is facilitated by the reduction of the development process into a series of individual phases. Since each phase is comprised of different tasks, different categories of developer on different pay rates can be employed, rather than having to pay for all-round ability.

An epistemological rationale was also identified as a role that a methodology can play. This relates to the gathering and systematizing of development knowledge. Methodologies provide a template for inexperienced developers, outlining the development stages and tasks that need to be performed. This was acknowledged as a factor in the interviews and received support from the survey respondents. The provision of a template for development helps to facilitate intercommunication and interchangeability among developers. Also, in relation to this epistemological role, an interviewee expressed the view that methodologies should play a role in helping organizations to learn from previous development projects.

5.6.2 Covert Political Roles

A number of political roles that methodologies can play were also identified in the study. First, the reductionist subdivision of the development process implicit in a methodology may help developers to cope with the stressful complexity of development. This study found that methodologies play a role as a *comfort factor*, suggesting that "proper" practices are being followed. This is related to the means-ends inversion phenomenon, identified in Table 2, to refer to the situation whereby developers can become so engrossed in blind adherence to methodologies that they lose sight of the fact that development of a system is the actual goal. One is reminded of Wittgenstein's dictum of not letting the method pass the problem by. Another similar role that a methodology can play is that of "legitimacy factor." In this scenario, organizations may claim to use a methodology to win contracts with government agencies or to help achieve ISO-certification. During the interviews, these roles were identified, with a number of organizations citing the fact that claiming to use a methodology was useful in marketing the organization's development process as having a certain standard of quality.

Also, some interviewees felt that methodologies helped to establish an aura of professionalism in relation to development—as one interviewee put it: "It provides a statement to the effect that we've got standards for our work, just like other departments." This could then be used to insulate IS departments from being forced to meet unreasonable deadlines for systems development from user departments. A related role identified was that of promoting the IS department to a more proactive role in planning and strategy formulation through building a business case to support development, rather than being consulted last in a reactive mode when company strategy had already been formulated. Thus, methodologies could be used to raise the profile of the IS department. Additionally, individuals within IS departments were able to use their position as methodology "champions" as a power base by which they could raise their profile within the IS department and the organization.

Other similar roles were also identified: for example, it was suggested that methodologies provide a *confidence factor* in justifying and supporting expensive investment decisions in development; that is to say, the systematic approach afforded by a methodology provided management with more confidence that the expenditure on a development project was justifiable. Also, by documenting all steps in the development process, methodologies can provide an *audit trail*, which illustrates the rationale behind decisions taken at various stages during development. This role is related to the *comfort factor* one described earlier.

6. CONCLUSIONS

The framework and the study in general lead to a number of implications, and these are discussed here.

The notion of a formalized development methodology seems not to be all that relevant in practice, in so far as a *methodology-in-action* appears to be uniquely enacted for each development project. Thus, a classification of systems development based on the use of specific formalized methodologies may not be all that useful in differentiating the development process in organizations. This is supported by the survey findings, where it transpired that the time allocated to development phases did not differ whether developers were following a formalized methodology or not. Also, support arises in the interview finding that even in those organizations that appeared to have based their development approach on the same commercial methodologies, very different methodologies had been compiled, in that they had customized their development approaches to the particular needs of their development environment. However, much was left to the discretion of individual developers as to what aspects to omit or include. Departures from the methodology are common; however, they are conscious and deliberate rather than arbitrary, evidence, perhaps, of maturity in relation to methodology usage. Thus, the methodology-in-action is one that has been heavily customized to the exigencies of the development situation, as certain aspects of the methodology may be omitted, while other phases may be supplemented, but in a deliberate fashion. Developers may have rejected a formalized commercial methodology, but perhaps for pragmatic reasons rather than due to ignorance as has been suggested in the literature.

The profile of the development environment is vastly different from that which prevailed when many of the currently available commercial methodologies were first mooted some 25 to 30 year ago. Thus, there is a need to 'advance the clock' by deriving sensible methodological canons more suited to the needs of the current development climate. For example, both integration and customization of packages and outsourcing are quite prevalent in today's environment (Fitzgerald 1998), yet few methodologies cater for these phenomena. Also, business systems development is often algorithmically simple. Thus, methodologies which may be strong in the area of real-time engineering systems design may not be as appropriate for business systems development. This is possibly even more relevant given the emergence of OO methodologies as many of the latter have been derived from experience with real-time applications, where object persistence—a fundamental feature of business applications where data storage is a key issue—is not necessary. Also, a recurring theme during interviews was that while methodologies could introduce rigor to the development process, productivity would necessarily suffer, and this trade-off was not tolerable. Thus, there may be a sense in which "good enough" systems can be developed in an appropriate time-scale, rather than striving toward delivering optimum solutions in an unreasonable time-scale.

Formalized methodologies may therefore be more placebo than panacea and provide a safety net for developers, recording the fact that "proper" practices were followed. However, as a result, developers have often been inclined to fall victim to goal displacement, following the methodology at the expense of actual systems development—a case of means-ends inversion. The quest for ISO certification in many organizations is also characteristic of this phenomenon.

As already mentioned, many researchers and practitioners continue to see the solution to systems development problems in terms of increased control and the more widespread adoption of formalized development methodologies (Fitzgerald 1996). The arguments and pressures that support the use of such methodologies, as summarized in Table 1, are indeed significant, but the problems associated with the use of methodologies, identified in Table 2, have not perhaps received adequate attention in the literature. Thus, the assumption that increased adoption of methodologies would help address the problems inherent in systems development is by no means proven.

References

Aaen, I. "Systems Development and Theory—In Search of Identification," in H. Nissen and G. Sandstrom (eds.), *Quality of Work versus Quality of Information Systems*, Lund University Press, Lund, Sweden, 1986, pp. 202-223.

Ahituv, N.; Hadass, M.; and Neumann, S. "A Flexible Approach to Information System Development," *MIS Quarterly* (8:2), 1984, pp. 69-78.

Argyris, C., and Schön, D. *Theory in Practice: Increasing Professional Effectiveness*, Jossey-Bass, San Francisco, 1974.

Avison, D., and Fitzgerald, G. *Information Systems Development: Methodologies, Techniques and Tools*, Blackwell Scientific Publications, Oxford, 1995.

Avison, D.; Fitzgerald, G.; and Wood-Harper, A. "Information Systems Development: A Tool Kit Is Not Enough," *The Computer Journal* (31:4), 1988, pp. 379-380.

Bansler, J., and Havn, E. "Information Systems Development with Generic Systems," in W. Baets (ed.), *Proceedings of Second European Conference on Information Systems*, Nijenrode University Press, Breukelen, The Netherlands, 1994, pp. 707-718.

Bantleman, J., and Jones, A. "Systems Analysis Methodologies: A Research Project," in T. Bemelmans (ed.), *Beyond Productivity: Information Systems Development for Organizational Effectiveness*, North Holland Press, Amsterdam, 1984, pp. 213-227.

Baskerville, R.; Travis, J.; and Truex, D. "Systems Without Method: the Impact of New Technologies on Information Systems Development Projects," in K. Kendall, K. Lyytinen, and J. DeGross (eds.), *The Impact of Computer Supported Technologies on Information Systems Development*, North Holland Press, Amsterdam, 1992, pp. 241-269.

Bodker, K., and Pedersen, J. "Workplace Cultures: Looking at Artefacts, Symbols and Practices," in J. Greenbaum and M. King (eds.), *Design at Work: Collaborative Design of Computer Systems*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1991, pp. 121-136.

Boehm, B. *Software Engineering Economics*, Prentice Hall, Englewood Cliffs, NJ, 1981.

Box, G. "Robustness in the Strategy of Scientific Model Building," in R. Launer and G. Wilkinson (eds.), *Robustness in Statistics*, Academic Press, New York, 1979, pp. 199-217.

Brooks, F. "No Silver Bullet: Essence and Accidents of Software Engineering," *IEEE Computer* (21:4), April 1987, pp. 10-19.

Brown, P. "Managing Software Development," *Datamation* (31:8), April 15, 1985, pp. 133-136.

Checkland, P. *Systems Thinking, Systems Practice*, Wiley, Chichester, UK, 1981.

Checkland, P. "Notes on Teaching and Researching IS," *Systemist (IS Special Edition Part II)* (16:1), 1994, pp. 6-8.

Chikofsky, E. "How to Lose Productivity with Productivity Tools," *Software Development: Computer-Aided Software Engineering*, IEEE Computer Society Press, Washington, DC, 1989, pp. 120-124.

Curtis, B.; Krasner, H.; and Iscoe, N. "A Field Study of the Software Design Process for Large Systems," *Communications of the ACM* (31:11), November 1988, pp. 1268-1287.

Davis, G. B., and Olson, M. H. *Management Information Systems: Conceptual Foundations, Structure and Development*, 2nd ed., McGraw-Hill Book Company, New York, 1985.

DeGrace, P., and Stahl, L. *Wicked Problems, Righteous Solutions: A Catalogue of Modern Software Engineering Paradigms*, Yourdon Press, Englewood Cliffs, NJ, 1990.

Downs, E.; Clare, P.; and Coe, I. *Structured Systems Analysis and Design Method: Application and Context*, Prentice-Hall International (UK), Hertfordshire, UK, 1992.

Fitzgerald, B. "Whither Systems Development: Time to Move the Lamppost?" in C. Lissoni, T. Richardson, R. Miles, A. T. Wood-Harper, and N. Jayaratna (eds.), *Proceedings of Second BCS Conference on Information Systems Methodologies*, BCS Publications, Swindon, UK, 1994, pp.371-380.

Fitzgerald, B. "A Descriptive Framework for Analyzing Problems in the Application of Systems Development Methodologies," in N. Jayaratna, R. Miles, Y. Merali, and S. Probert (eds.), *Proceedings of Third Conference on Information Systems Methodologies*, BCS Publications, Swindon, UK, 1995, pp. 27-38.

Fitzgerald, B. "Formalized Systems Development Methodologies: A Critical Perspective," *Information Systems Journal* (6:1), 1996, pp. 3-23.

Fitzgerald, B. "The Use of Systems Development Methodologies in Practice: A Field Study," *Information Systems Journal* (7:3), 1997, pp. 201-212.

Fitzgerald, B. "An Empirical Investigation into the Adoption of Systems Development Methodologies," *Information & Management*, 1998 (forthcoming).

Floyd, C. "Outline of a Paradigm Change in Software Engineering," inG. Bjerknes, P. Ehn, and M. King (eds.), *Computers and Democracy: A Scandinavian Challenge*, Avebury Gower, Brookfield, VT, 1987, pp. 147-163.

Folkes, S., and Stubenvoll, S. *Accelerated Systems Development*, Prentice Hall, London, 1992.

Friedman, A. *Computer Systems Development: History, Organization and Implementation*, Wiley & Sons, Chichester, UK, 1989.

Holloway, S. *Methodology Handbook for Information Managers*, Gower Technical, Aldershot, UK, 1989.

Hough, D. "Rapid Delivery: An Evolutionary Approach for Application Development," *IBM Systems Journal* (32:3), 1993, pp. 397-419.

Jayaratna, N. *Understanding and Evaluating Methodologies*, McGraw-Hill, London, 1994.

Jenkins, A.; Naumann, J;. and Wetherbe, J. "Empirical Investigation of Systems Development Practices and Results," *Information & Management* (7:3), 1984, pp. 73-82.

Lee, A. "Integrating Positivist and Interpretivist Approaches to Organizational Research," *Organizational Science* (2:4), November, 1991, pp. 342-365.

Lyytinen, K. "A Taxonomic Perspective on Information Systems Development," in R. Boland and R. Hirschheim (eds.), *Critical Issues in Information Systems Research*, John Wiley and Sons, Chichester, UK, 1987, pp. 3-41.

Miles, M., and Huberman, A. *Qualitative Data Analysis: A Sourcebook of New Methods*, Sage, Beverley Hills, CA, 1984.

Olerup, A. "Design Approaches: A Comparative Study of Information System Design and Architectural Design," *The Computer Journal* (34:3), 1991, pp. 215-224.

Orlikowski, W. "Integrated Information Environment or Matrix of Control? The Contradictory Implications of Information Technology," *Accounting, Management and Information Technolgies* (1:1), 1991, pp. 9-42.

Page-Jones, M. "Structured Methods are Dead: Long Live Structured Methods," *American Programmer*, November 1991, pp. 31-37.

Peters, L., and Tripp, L. "Comparing Software Design Methodologies," *Datamation*, November 1977, pp. 89-94.

Stage, J. "The Use of Descriptions in the Analysis and Design of Information Systems," In R. Stamper, P. Kerola, R. Lee, and K. Lyytinen (eds), *Collaborative Work, Social Communications and Information Systems*, North Holland, Amsterdam, 1991, pp. 237-260.

Stolterman, E. "The 'Transfer of Rationality,' Acceptability, Adaptability and Transparency of Methods," in W. Baets (ed.), *Proceedings of Second European Conference on Information Systems*, Nijenrode University Press, Breukelen, The Netherlands, 1994, pp. 533-540.

Vitalari, N., and Dickson, G. "Problem Solving for Effective Systems Analysis: An Experimental Exploration," *Communications of the ACM* (26:11), November 1983, pp. 948-956.

Ward, P. "The Evolution of Structured Analysis: Part I—The Early Years," *American Programmer* (4:11), 1991, pp. 4-16.

Wastell, D., and Newman, M. "The Behavioral Dynamics of Information Systems Development: A Stress Perspective," *Accounting, Management & Information Technology* (3:2), 1993, pp. 121-148.