Please cite as: Fitzgerald, B. (1997) The Use of Systems Development Methodologies in Practice: A Field Study, *The Information Systems Journal*, Vol. 7, No. 3, pp. 201-212.

The Use of Systems Development Methodologies in Practice: A Field Study

Abstract

This paper discusses the findings of a field study of experienced systems developers which investigated the usage of systems development methodologies (SDMs) in practice. Among the main findings to emerge are the following: There is a wide difference between the formalised sequence of steps and stages prescribed by a methodology, and the methodology-in-action uniquely enacted for each development project. Also, there is much evidence which suggests that developers omit certain aspects of methodologies, not from a position of ignorance, but from the more pragmatic basis that certain aspects are not relevant to the development environment they face. Thus, methodologies are tailored to the needs of the prevailing development environment, and, being documented in a single manual often, are framed at a higher level of granularity which specify broad activities and objectives, rather than specifying in a very detailed fashion the exact manner in which development should take place. Additionally, the findings clarify the apparently contradictory position in the literature as to whether developer experience is positively or negatively correlated with methodology usage. The study would suggest that experienced developers are likely to use methodologies, albeit ones which have been heavily customised to the exigenceis of the particular situation.

1 INTRODUCTION

Ultimately, systems development methodologies should serve to benefit developers to some extent. Thus, it is not surprising that an examination of the literature reveals a general bias in favour of methodologies (cf. Fitzgerald, 1996a), with an implicit assumption often that methodologies should be applied rigorously and in totality (Humphrey *et al*., 1991). The fact that many practitioners have not so far adopted methodologies has been explained variously as either due to ignorance among practitioners (Ward, 1991) or the slowness of technology transfer (Chikofsky, 1989). However, researchers have criticised the lack of empirical research on systems development in real organisational contexts, and there have been calls for a "clearer understanding of the realities of software development" (CSTB, 1990). As McLean (1973) aptly put it: "the proper place to study elephants is the jungle, not the zoo". More research is therefore needed into the actual practice of systems development in organisations, justifiable even solely on the basis that practice has often preceded theory in the field.

The research study reported in this paper was concerned with investigating the practice of systems development and the nature of methodology usage in some depth. It involved a number of field interviews which sought to complement a postal survey of the use of methodologies which had been carried out earlier (cf. Fitzgerald, 1996b). Among the survey findings were that 60 percent of respondents were not using formalised methodologies. Also, even in those organisations who reported using the same methodologies, the survey findings suggested that there was little convergence in development practices. Thus, the survey lent support to the view that a unique methodology-in-action is enacted for each development project. This issue was explored in the study reported here. Also, the survey findings suggested that ignorance might not be a useful explanatory factor in the non-use of methodologies, as the organisations who were not using them still had a high degree of knowledge about them. This was also explored in this study. Additionally, the survey findings suggested that inexperienced developers were more likely to follow a methodology rigorously. This finding would support that of Lee and Kim (1992) and Kozar (1989). However, Leonard-Barton's (1987) research would suggest that experienced developers are more likely to use a methodology, as they would be aware of the benfits, and indeed, this is a view in accordance with the positive bias in the literature. This issue was also explored in this study.

2 RESEARCH METHOD

In the case of postal surveys, random sampling strategies are often desirable since statistical analysis is usually an integral feature. However, in case-based research, given that sample sizes cannot be as large, simple random selection is not appropriate. Researchers have stressed the importance of a theoretical or purposive strategy (Eisenhardt, 1989, Pettigrew, 1990). The basic rationale for such a strategy is that given the small number of organisations that can realistically be included, one needs to use a selection strategy that can serve as a lens to magnify the research topic. Thus, Pettigrew (1990) argues for the inclusion of polar opposites in the sample, as this provides extreme situations to contrast the topic of interest. In this study, companies were included from across the spectrum of methodology usage which comprises four categories, viz., those using a formalised commercial methodology, those using an internal methodology based on a formalised commercial one, those using an internal methodology not based on a formalised commercial one, those not using any formalised methodology.

Patton (1990) recommends an intensity sampling strategy whereby cases which represent categories of extreme interest are focused on in more detail. This was addressed in this study by selecting a number of companies who were not using a formalised methodology, and a number of companies who were using formalised commercial methodologies.

This selection strategy yielded eight companies in total. Two people were interviewed in each organisation, from IS management through to actual hands-on development level. Each interviewee was subsequently provided with a write-up of the interview findings and their opinion on the researcher's interpretation was elicited. One written and three telephone responses were received, and all expressed satisfaction with the validity of the researcher's interpretation. Table 1 provides summary details on the organisations in which the personal interviews took place.

Company	**Business sector**	**No. of emp-loyees**	**No. in IS**	**No. of developers on typical project**	**No. of months on typical project**	**Methodology recommended in company**
Allied Irish Banks (AIB)	Financial Services	16000	600	10-12	20	Information Engineering
Dept. of Education	Govt Dept.	950	15	4	6	SSADM
Core Computing Ltd.	Software House	52	43	5	3 - 6	Internal based on Oracle*Case & SSADM
Naval Service: Dept. of Defence	Govt Dept.	1200	10	4 - 6	6 - 15	Internal based on Oracle*Case & SSADM
Dairygold Co-op. Ltd.	Food Co-operative	2500	20	1 - 2	1 - 3	Internal not based on commercial SDM
Bord Gais Eireann (BGE)	Energy Service Provider	800	30	5	30	Not using a formalised SDM
Trustee Savings Bank (TSB)	Financial Services	1200	40	2	1.5	Not using a formalised SDM
Pfizer Pharma-ceuticals	Manufact-uring	250	6	2	3	Not using a formalised SDM

Table 1 *Organisational Background Information*

A qualitative approach to data analysis was adopted in the study. Researchers have suggested that when using a qualitative approach, a set of initial *seed* categories may be generated to guide the research (Miles & Huberman, 1984). Thus, the survey findings (Fitzgerald, 1996b), coupled with the author’s commercial experience (12 years) as a systems developer were drawn upon to generate a number of seed categories. These categories are summarised in Table 2.

As already mentioned, two people were interviewed in each organisation—one from an IS management background, and one with a more direct hands-on development role, i.e., project leader, analyst or programmer. The duration of interviews ranged between 30 minutes and 90 minutes. Interviews were recorded so as to minimise data loss due to note-taking, and these recordings were subsequently coded according to the seed categories presented in Table 2.

Seed Categories	***Illustrative Questions***
Formalised methodology v. methodology in action: A distinction may be drawn between the original methodology as formalised in a manual and the methodology-in-action as interpreted by the developer. This was evident in the postal survey (Fitzgerald, 1996b) where only 6% of respondents stated that they were rigorously following formalised commercial methodologies	Interviewee's definition of methodology? Tailoring of methodology for each project? What aspects are omitted? On what basis? Tailoring of methodology for each project? Extent to which methodology is rigorously followed How is it documented?
Methodology usage: The survey findings (Fitzgerald, 1996b) showed that 60% of respondents were not using any formalised methodology. The findings also suggested that inexperienced developers were more likely to use methodologies than experienced ones.	If methodology not used: Why not? What are characteristics of approach? Does developer experience make methodology usage more or less likely?

Table 2 *Initial Seed Categories*

3 ANALYSIS OF INTERVIEW FINDINGS

3.1 FORMALISED METHODOLOGY V.METHODOLOGY-IN-ACTION

The interviews confirm the view of a methodology-in-action, uniquely enacted for each development project. Evidence to support this interpretation comes from a number of factors. Firstly, *none* of the organisations who were using methodologies were following them rigorously. However, several interviewees stressed the fact that when they departed from the methodology, they did so consciously and were very aware of the implications of such an action. Also, several organisations had compiled their own methodologies which they had documented in one or two manuals, specifically tailored to the contingencies of their development situation. These tailored methodologies tended to emphasise those subsets of the development lifecycle that were most relevant to them. Also, being cast as a single manual often, they were framed at a higher level of granularity, rather than prescribing the minute details of each development step and how it should be performed. These issues are discussed below.

Methodologies not being followed rigorously

As already mentioned, none of the organisations who were using methodologies were following them rigorously. This was all the more significant given the fact that some of these organisations had specifically tuned their methodologies to the specific contingencies of their development situation. As one interviewee in the Naval Service put it:

> "Obviously, it [the methodology] is not followed rigorously. That would lead to far too much paperwork"

In Allied Irish Banks (AIB), where the Information Engineering methodology was recommended, it transpired that the methodology had been used on a limited number of projects, and was heavily customised to the needs of the situation, as there were "a lot of things outside the methodology", according to the IS manager. Similarly, the project leader interviewed in the Department of Education, where SSADM was recommended, admitted that the actual use of SSADM was "very patchy" in the Department.

Conscious departure from methodology

Core Computing were one of the organisations in which the methodology was not always followed. The IS manager explained that given the time-constraints on certain development projects, some aspects of development specified by the methodology might have to be omitted, system documentation, for example. He believed that "departure from the methodology is inevitable in the real world". However, he identified as "critical" the fact that it was a "conscious and deliberate" departure. He stated that the client would be informed if necessary. However, extra care and vigilance would be taken to ensure that these omitted aspects would eventually be completed to comply with methodological requirements.

A similar sentiment was also expressed by the IS manager in the Naval Service. He readily acknowledged that their methodology was not followed rigorously, but he identified the consequent need to “watch out down the line”, saying that one “could go to war on it now, or let it slip and move on”.

In AIB, one of the interviewees indicated that for development on legacy systems, the recommended methodology, Information Engineering, was not used, rather an internally-derived methodology was used. This interviewee expressed the view that it would be extremely difficult to use new methodologies such as Information Engineering when upgrading or bridging legacy systems, as the basic structure and documentation of the legacy systems would not be an adequate basis upon which to apply a methodology such as Information Engineering. (This is perhaps contrary to the views of the vendors of the Information Engineering methodology). In a similar vein in the Dept. of Education, the project leader stated that there weren’t any “rigorous users of SSADM as the payback [was] not there”.

Thus, in these companies, methodology non-usage was not treated lightly or in an arbitrary fashion, and certainly ignorance was not a factor, as has been suggested in the literature (Ward, 1991). There was a high level of pragmatic consideration given to the matter, and the need to be adaptive depending on the contingencies of the specific situation was widely recognised.

Methodology tailoring: salient subsets of lifecycle

As already mentioned, several organisations had created a tailored methodology suited to the specific needs of their development environment. Core Computing, for example, who had been classified as using a commercial methodology following the survey phase, had actually compiled their own methodology which was documented in a single manual. This methodology was described as based on two commercial methodologies, SSADM and Oracle*Case (of which they had bought the entire suite), and contained high-level guidelines on the development lifecycle activities most salient to the needs of their development environment. Thus, given that Core Computing were a software house, their methodology manual focused on issues such as system testing (identified as important on several occasions by both interviewees in this organisation), documentation, configuration management and telephone support. These issues are especially important in a software house which has to maintain several versions of several packages across multiple customer sites. However, most methodologies do not consider such downstream activities as telephone support. The IS manager, who stated that he himself was not a disciplinarian, was strongly of the opinion that disciplined work practices were critical in a software house. However, he recognised that there were trade-offs between development productivity and the rigour that a methodology imposes, but he argued that it was vital that an organisation should:

> "fine-tune the methodology to suit their particular needs, otherwise it's a case of singing from someone else's hymn sheet".

The Naval Service had also derived their own formalised methodology, drawing on the same two commercial methodologies as Core Computing, viz., SSADM and Oracle*Case. Again, it was documented in a single manual. However, their methodology, even though based on the same commercial methodologies, differed greatly from that created by Core Computing. The Naval Service methodology focused on issues relevant to them and was very prescriptive in terms of project initiation, for example, establishing clear and formal guidelines for preparing specifications for development which should be sent to tender. As can be seen from Table 2, even though the organisation were large in terms of numbers employed (1200), they were served by quite a small IS department (10 people). Thus, joint development, whereby development was done jointly between in-house developers and outside contractors was a common mode of development. The Naval Service were very systematic in their approach to development projects which could typically begin with a formal Request for Information (RFI) from potential vendors/suppliers. This would then be followed by a formal Request for Proposal (RFP), and following this a project steering committee would be established. Thus, a formalised approach was suited to this type of environment.

However, an interesting feature of the Naval Service was that career officers would often spend some part of their career gaining experience in the IS department, but would not remain there on a permanent basis. Thus, one interviewee identified the usefulness of a formalised methodology in providing a "template" to inform those who were new to the area of "what to do and where to go next".

In AIB, Information Engineering was only being used in the early planning and analysis phases of development. The IS manager stressed the need for systems development to adopt a more business strategy-driven focus (this rationale will be discussed in a subsequent section). However, the IS manager stated that they found a need to supplement the methodology with a number of additional stages and techniques, citing the area of cost-benefit analysis as an example, where they used a number of sophisticated techniques which were not available as part of the methodology.

Dairygold, as has already been mentioned, were in the process of adopting a formalised methodology. However, they were not going to use a commercial one. They had evaluated SSADM but rejected it as "incredibly cumbersome", and did not see it as appropriate to their needs. This organisation were the most zealous in seeking to establish a formalised methodology. However, as already mentioned, part of the rationale was the desire to achieve ISO certification. Thus, development and documentation standards, which were key to ISO-certification, were to the forefront in Dairygold.

TSB indicated that they were not using a formalised methodology. Notwithstanding this, they had produced a manual which summarised the main requirements for development in the type of fourth generation environment that they identified as prevailing in TSB. Analysis of business requirements and system testing were both seen as vital in this organisation, and they had established two distinct internal consultancy services within the organisation which specialised in these areas (although unit testing was still done by the IS department). Thus, these phases of development were actually formalised outside the IS department and were not explicitly covered by the methodology. Within the IS department, an interviewee characterised the situation as "development just takes place". Their manual focused on "coding standards, reuse of code, and ways of approaching problems". However, this organisation's productivity was extremely impressive. For example, following a recent merger between TSB and a sister bank, they had rewritten an entire suite of banking systems within eight months. Likewise, when the two major banks in the country announced that they would be introducing a charge card service within 5 months, TSB were the only other bank who could develop a system for this new service within such a time-scale.

The project leader interviewed in Bord Gais Eireann (BGE) indicated that, while they did not follow a formalised methodology, they did have a "definite data-driven approach underpinned by a Data Dictionary—our Bible", as he termed it. They had compiled two manuals to support development—a programming standards manual and a database standards manual. Again, the project leader in BGE felt that their fourth generation environment was very different to a "COBOL environment" where he believed formalised methodologies were more applicable.

Single manual: larger granularity

The fact that three of the organisations had compiled a single methodology manual which was framed at a higher level of granularity, containing high level guidelines, suggests that the multiplicity of manuals which accompany many methodologies and prescribe in a very detailed fashion the exact manner in which development should take place is not suited to the actual needs of developers in practice. Thus, methodologies were uniquely enacted for each development situation, but within broad parameters. For example, the IS manager in Core Computing suggested that "a methodology should define standards and procedures where appropriate", but acknowledged that they should be used in "a similar way and not necessarily the same way" by different developers. He proposed the analogy of a journey to a particular destination, suggesting that there would be towns that one should pass through on the way, but issues relating to where to stop, speed and mode of transport could vary from individual to individual.

In the Naval Service, as already mentioned, their methodology was derived from the same commercial methodologies as Core Computing. It provided clear guidelines on early development phases and less on actual program design and coding, as these were often outsourced under joint development. Documentation was also considered important. However, the IS manager characterised the methodology as signifying:

> "measurable statements of intent...even if not everyone is tuned in to it, it provides signposts of the general factors that should be considered at least".

3.2 FACTORS RELEVANT TO METHODOLOGY USAGE AND NON-USAGE

Two issues which arise in the literature in relation to methodology usage are the following: Firstly, whether lack of use is due primarily to ignorance, as has been suggested (Ward, 1991), and secondly, whether more experienced developers are more likely to use methodologies (Leonard-Barton, 1987) or less likely to use them (Kozar, 1989; Lee & Kim, 1992). These issues were explored further in this phase of the research, and are discussed next.

Non-usage of methodology: ignorance or atrophy due to lack of relevance?

The evidence from the interviews suggests that ignorance is not an explanatory factor in the organisations who do not use commercial methodologies. Core Computing and the Naval Service had created their own tailored methodologies following experience with commercial methodologies, which would suggest that the decision not to use them was based on a position of knowledge, rather than ignorance. In the Department of Education, the project leader had experience of strict adherence to a methodology in a previous organisation, and expressed the view that development had been "much worse" in that organisation. As already mentioned, Dairygold had evaluated SSADM, but had rejected it as "incredibly cumbersome" and not suited to the needs of their environment.

In the organisations who were not using formalised methodologies, this issue is obviously of greater significance. However, again, ignorance does not appear to be an explanatory factor. TSB, for example, had evaluated several commercial methodologies in the past but had concluded that they were "more applicable to a third generation development environment" and not suited to the type of fourth generation environment that prevailed in TSB. In BGE, the project leader had used an internal formalised methodology in a previous job, but did not believe the particular methodology would have been appropriate in BGE. They had also evaluated both SSADM and PRINCE but had concluded that they were not suited to their development environment, as the "payback was not worth it". In Pfizer Pharmaceuticals, who were the least formalised in their development approach, development was primarily concerned with small add-ons to existing systems and integration of packages. The IS manager felt that a "formalised methodology would be a bit restraining".

Several interviewees were uncomfortable when asked to provide a definition of a development methodology. There was a tendency to view methodologies in fairly simplistic terms often, equating them with naming conventions or techniques such as data flow diagramming. This might appear at first inspection to indicate ignorance. However, these interviewees were also often unable to remember the name of particular development techniques, such as entity relationship modelling, process mini-specifications, and the like. However, they had once been familiar with these techniques but their knowledge had atrophied due to lack of use and relevance to their situation. Likewise, the interviewee in TSB characterised the situation well, saying that "development just takes place". Thus, for many interviewees, methodologies were not all that relevant to their everyday work, and knowledge and consideration of definitions had atrophied. This was reinforced by the IS manager in Core Computing who stated that “developers don’t tend to refer back to methodologies once development is underway”.

Developer experience and methodology usage?

As already mentioned, there are conflicting views in the literature as to whether development experience is positively correlated with methodology usage. The argument in favour of positive correlation suggests that experienced developers will realise the benefits that a methodology provides and will use it. The argument against suggests that inexperienced developers will use a methodology due to uncertainty, whereas experienced developers will find a methodology restraining and are more likely to step outside it. This latter argument received some support during these interviews. For example, the IS manager in Dairygold expressed the view that “experienced developers find a methodology a bit of a hindrance”. Likewise a developer in Pfizer Pharmaceuticals stated that he had used specification forms as a junior programmer, but felt that “more experienced developers would be more likely to take a flier”. This view was reiterated by the IS manager in the company who stated that “experienced developers could be completely fettered by methodologies”. The IS manager in Core Computing suggested that new developers are now very amenable to following formalised methodologies as they may have been exposed to them as part of their university or college training in contrast to developers in the past, many of whom acquired their development skills through practice, without formal education.

From this evidence, one might conclude that methodology usage is negatively correlated with developer experience. However, methodologies were being used by experienced developers in the organisations studied. As already mentioned, developers in Core Computing did depart from the methodology, but in a conscious and deliberate fashion. The IS manager suggested that experienced developers do not object to methodological guidelines that make sense. This theme was echoed in both Dairygold and the Naval Service. As already mentioned, Dairygold were in the process of implementing their own formalised methodology. The project manager in charge of implementation indicated that he did not consider there to be resistance from experienced developers to the methodology; rather, he described it as a “challenge” from developers, who initially questioned the proposed methodology, but, once persuaded of the benefits of the methodology, were happy to accept it.

In the Naval Service, the IS manager described a similar scenario in relation to methodology usage. He felt that inexperienced developers would follow one because of uncertainty or lack of confidence. However, once they gained some experience, they tended to take more chances and follow an informal development approach. However, they eventually came to realise the negative implications of not following certain aspects of the methodology, and came around to increased methodology usage. Likewise in TSB, the IS manager indicated that experienced developers would use a methodology if it was logical and made sense:

> "A sensible methodology will be used by experienced developers. They know that they have to clean up the dirt later"

This is in keeping with the view expressed by an interviewee in BGE who described their development approach as just another word for "common-sense", suggesting that it had been compiled from "knowledge that was in the air".

Thus, based on evidence from the interviews, a U-shaped curve seems to capture the relationship between developer experience and methodology usage, as depicted in Fig. 1. However, the interview evidence would suggest some mediating factors or pressures. Firstly, educational exposure to methodologies might predispose inexperienced developers more towards a formalised methodological approach. Also, inexperienced developers could find that a methodology provides a useful template for the development process. Subsequently, the realisation of the futility of blind and slavish adherence to low-level standards, prescribed as universally applicable, could cause developers to make less use of methodologies. Eventually, the derivation of a tailored methodology which provides sensible standards and guidelines at the appropriate level of granularity could predispose experienced developers towards increased methodology usage.

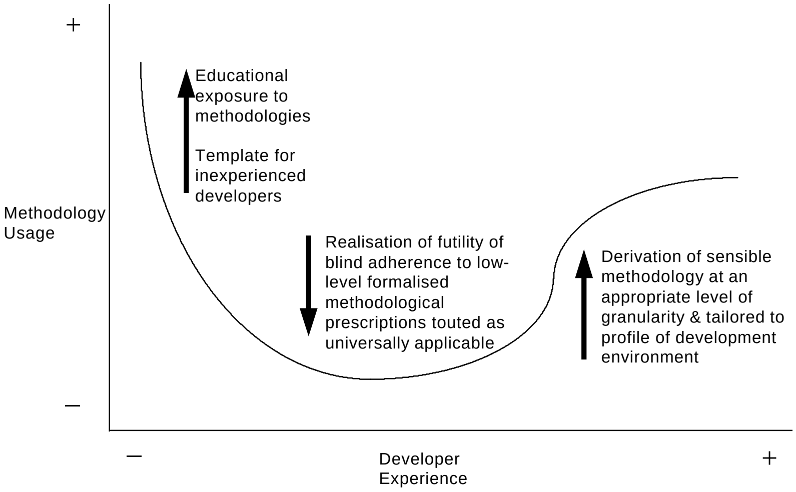
+
–
Methodology Usage
Educational exposure to methodologies
Template for inexperienced developers
Realisation of futility of blind adherence to low-level formalised methodological prescriptions touted as universally applicable
Derivation of sensible methodology at an appropriate level of granularity & tailored to profile of development environment
–
Developer Experience
+

Fig. 1 *Relationship Between Developer Experience and Methodology Usage*

5 CONCLUSION

The general implication in much of the literature is that practitioners are moving, albeit slowly, towards more widespread adoption of formalised methodologies, and that axiomatically, this adoption would improve both the process and product of systems development (Fitzgerald, 1996a). However, the findings of this study suggest that practitioners will not adopt formalised methodologies in their prescribed form, and, indeed, that they may be modifying and omitting aspects of methodologies in a very pragmatic and knowledgeable fashion. In fact, there was considerable evidence that methodologies are tailored quite precisely to the exigencies of the development environment faced in organisations currently. Also, it appears that these tailored methodologies are more likely to be adopted by experienced developers than generic methodologies which are touted as universally applicable. The findings serve to question whether there is any such thing as a formalised, commercial methodology, since even the organisations who claimed to use them, didn't in fact use them rigorously or in totality.

Some researchers have suggested that systems development may perhaps be beyond methodology (Wastell, 1996), but it is perhaps the case that the process was never well enough understood to definitively prescribe a methodology. Professions tend to develop in three broad stages. Initially, the profession may be something of an art form, and practitioners develop the skills needed to solve practical problems. At this stage there are enormous variations in skill levels. The next stage involves the codification of the skills of the best practitioners, and in the final stage, that of rationalisation, these codified skills are integrated into a formal curriculum where they can be taught. However, this progression has typically taken centuries rather than decades, thus the prescription of universally-applicable development methodologies may be wildly optimistic given the relative infancy of the IS field.

References

Chikofsky, E. (1989) How to lose productivity with productivity tools. *Proceedings of 3rd IFAC/IFIP Workshop*, IFAC Publications, Indiana, US, 1-4.

CSTB. (1990) Scaling up: a research agenda for software engineering, Communications of the ACM, Vol. 33, No. 3, 281-293.

Eisenhardt, K. (1989) Building theory from case study research, Academy of Management Review, 14, 4, 532-550.

Fitzgerald, B. (1996a), Formalised systems development methodologies: a critical perspective, Information Systems Journal, 6, 1, 3-23.

Fitzgerald, B. (1996b), An investigation of the use of systems development methodologies in practice, in Coelho, J., Jelassi, T., Konig, W. Krcmar, H., O'Callaghan, R. and Saaksjarvi, M. (Eds) Proceedings of the 4th European Conference on Information Systems, Ficha Tecnica, Lisbon, 143-162.

Humphrey, W., Snyder, T. and Willis, R. (1991) Software process improvement at Hughes Aircraft. *IEEE Software,* **July**, 11-23.

Kozar, K. (1989) Adopting systems development methods: an exploratory study. Journal of Management Information Systems, 5, 4, 73-86.

Lee, J. and Kim, S. (1992) The relationship bteween procedural formalisation and MIS success. Information & Management, 22, 89-111.

Leonard-Barton, D. (1987) Implementing structured software methodologies: a case of innovation in process technology. Interfaces, 17, 3, 6-17.

McLean, E. (1973) Comment on Empirical studies of management information systems, by Van Horn, R. Data Base, Vol. 4, No. 4, p.181.

Marshall, C. and Rossman, G. (1989) Designing Qualitative Research, Sage Publications, Newbury Park, California.

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

Patton, M. (1990) Qualitative Evaluation and Research Methods (2nd Edition), Sage Publications, London.

Pettigrew, A. (1990) Longitudinal field research on change: theory and practice, Organizational Science, 1, 3, 267-292.

Ward, P. (1991) The evolution of structured analysis: Part I--the early years. American Programmer, 4, 11, 4-16.

Wastell, D. (1996) The fetish of technique: methodology as a social defence, Information Systems Journal, 6, 1, 25-40.