GROUNDED THEORISING APPLIED TO IS RESEARCH — DEVELOPING A CODING STRATEGY

Bruce H. Rowlands
School of Computing & Information Technology
Griffith University
b.rowlands@griffith.edu.au

ABSTRACT

This paper provides an example of developing a coding strategy to build theory of the roles of methods in IS development. The research seeks to identify and understand how system development methods are used in an IS department within a large Australian bank. The paper details a theoretical framework, particulars of data collection, and documents an early phase of analysis – data reduction and the generation of an initial coding scheme. Guided by a framework to study the use of methods, the analysis demonstrates the framework's plausibility in order to develop theoretical relationships with which to develop a grounded theory.

Key Words— Grounded Theorising, coding schemes, systems development methods, interpretive case research.

INTRODUCTION

This paper contributes to the discussion about developing a coding strategy for qualitative research in Information Systems. The paper provides an example of developing a coding strategy to build theory of information systems development. The approach to building theory makes use of a method-in-action framework by demonstrating its plausibility to illuminate an under-researched aspect of information systems practice – how Systems Development Methodologies (SDMs) are used in organisations.

To contribute to the scientific documentation of method use, this paper provides aspects of a grounded study of the social and organisational issues surrounding the use of development methods in a large Australian financial institution. This paper addresses methodical issues about the conduct of using Grounded Theory (GT) for inductively generating theory. While having its antecedents in sociology in the mid 60s, in the field of IS there has been a growing application of GT, but limited methodological guidance for builders of theories, and few exemplars of conducted research. Therefore it is contended that there remains a paucity of published accounts of the application of GT, and in particular – approaches to coding in the field of IS. This paper makes a contribution to such a deficiency.

The remainder of the paper is organised as follows. Firstly, based on a review of the literature, the paper defines the features of a systems development methodology (SDM) and makes a case for considering the social and organisational context of method use. Second, guided by the work of Fitzgerald *et al* (2002:12), a framework is presented allowing focus on practitioner perceptions of the roles of development methods. Next, the research design, data collection and data analysis is discussed. The *Research Procedures* section discusses alternate approaches to coding with GT and provides more details of the 'how to' of coding and grounded theorising. The paper ends with a positive conclusion *vis a vis* the plausibility of the framework to illuminate an ISD practice, and also a call for further research in this area.

A REVIEW OF RELATED RESEARCH ON SYSTEMS DEVELOPMENT METHODS

For the purpose of this paper, a systems development methodology (SDM), as defined by

Iivari *et al* (1998), is a codified set of goal-oriented procedures which are intended to guide the work and cooperation of the various parties (stakeholders) involved in the building of an information systems application. Typically these procedures are supported by a set of preferred techniques and tools, and guiding principles. Furthermore, the terms *methodology* and *methods* are used inter-changeably and have become blurred by common usage, and now seem largely fused concepts in information systems parlance.

SDMs are promoted as a means of improving the end product of the software development process by specifying the activities and the documentation to be produced (Avison & Fitzgerald, 2003:20). The alleged benefits are standardisation of the process and consistency from one development project to the next. The assumption of desirability has been reflected in research that has emphasised methodology design, the comparison of technical features, and methodology selection. However, the research literature that emerged in the 1990s suggests that methodologies have not always been used, and if used, not always in the way they were intended (Fitzgerald *et al*, 2002:100). There is a tension between the technical intent of methodologies (or methods) and how methods are actually used. Recent empirical evidence suggests that methodologies are not used for technical reasons alone, but are in practice affected by organisational and behavioural factors (Sauer & Lau, 1997; Fitzgerald 1998; Truex *et al*, 2000;).

In research addressing the issue of methodology use, Fitzgerald, Russo & Stolterman (2002:12) proposed a framework to investigate the complex nature of systems development and the use of methods in practice. The framework has been constructed on the basis of detailed investigation of the systems development literature, and empirical research (Fitzgerald, 1997, 1998; Russo & Stolterman, 2000; Wynekoop & Russo, 1993, 1995, 1997). Despite the identified divergence between the intent of methodologies and concrete practice, researchers still have very little descriptive understanding of how practitioners use methodologies in their day-to-day work. The work that has been published is limited in its ability to consider the complex social and organisational context of methodology use. For instance, Wynekoop & Russo (1997) in a survey of the existing literature on SDMs revealed that over half of the 123 research papers examined consisted of normative research in which concept development was not based on any empirical grounding, but merely on the authors' speculations or opinions. Most field research on methodology use (with the noted exception of Madsen & Kautz, 2002; Fitzgerald *et al*, 2003; Kiely & Fitzgerald, 2003; Hansen et al, 2003; and Vidgen *et al*, 2004) has a method engineering and tool orientation that brings focus to structural aspects of the methodology, but without any consideration of organisational issues. Few studies have been conducted in order to identify how SDMs are selected or adapted, or how they are used. There also appears to be few practice demonstrations or case studies illustrating the method-in-action process with the noted exception of Fitzgerald et al (2003) and Vidgen et al (2004). Clearly, the lack of process-oriented research in this area precludes a full understanding of how SDMs are used. Researchers have long called for research on methodologies in real life organisational situations (Iivari & Maansaari, 1998) and they continue to do so (Beynon-Davies & Williams, 2003). With the call in mind, and in the absence of a validated model of methodology use, the paper applies the Fitzgerald *et al* (2002) method-in-action framework as a conceptual lens to understand methodology use within a single organisational setting. The framework is depicted in Figure 1, and each element is explained briefly in the next section.

A FRAMEWORK FOR CONCEPTUALISING METHODOLOGY USE

This section examines the usefulness of a framework for studying method use. Most qualitative researchers attempt to avoid prior commitment to theoretical constructs before gathering any data. Yet, as discussed by Whetten (1989), two different approaches may be taken, or combined. In the first, the researcher works within an explicit theoretical framework. Therefore, a theoretical framework becomes a researcher's first cut at making some explicit theoretical statements (Miles & Huberman, 1994:18). This approach is known as deduction. In the second, the researcher tries not to be constrained by prior theory and instead sees the development of relevant theory, propositions, and concepts as a purpose of the project. This approach is generally known as induction.

In this example, both approaches were combined since the main intent was to study a relatively un-researched topic (the enactment of SDMs in development projects) within the bounds of an already well-established research program (process of software development). While it is possible to be inductive, I chose not to ignore previous work in the field, and instead adopted a loose conceptual model built on previous research that was a conceptual advance on the literature. The model of method use, as proposed by Fitzgerald *et al* (2002), is comprised of six sets of issues. These issues were used to develop the initial coding scheme for the qualitative analysis of data (to be discussed in the *Research Procedures* section. The pivotal component of the model, the method-in-action process, is to be developed and presented in *future research*, and does not form part of this paper's discussion. However, and this is a most important point, given that this study was aimed at theory building, not theory testing, the theoretical framework and model was used solely as a guide. It helped make sense of what occurred in the field, ensured that important issues were not overlooked, provided a set of tentative constructs to be investigated, and guided my interpretation and focus.

The framework is briefly described in the following paragraphs.

Systems Development Methodologies can be commercial (sold or recognised outside a single organisation), home-grown (developed and used within a single organisation), or adapted (that is, tailored to suit the local contingencies of the task at hand).

Roles of Method include two broad, but diametrically opposed categories of roles that methods can play in the development process. First, there are a set of rational or intellectual roles forming part of the conceptual basis and rationale behind the use of methods. These roles include, for example, the reduction of complexity of systems development, facilitation of project management & control, the division of labour, systematisation of development knowledge, and standardisation of the development process. These roles are countered (as alleged by Fitzgerald *et al*, 2002:103) by a set of political roles that are more covert in nature. These roles include: professionalising IS development work, help make IS development more proactive in strategy formulation, a comfort/confidence factor, an audit trail, legitimacy factor, and a power base for a methodology champion.

Method-in-Action is the actual use of the methodology and can be uniquely enacted by the developers. The unique enactment is reflected in the framework in Fig 1 by the fuzzy, cloudy outlines.

Development Context is represented by the cloud shape in Fig 1, reflecting the fact that development and maintenance takes place in a unique real context. This context cannot be easily analysed to produce a neat and regular specification of user requirements.

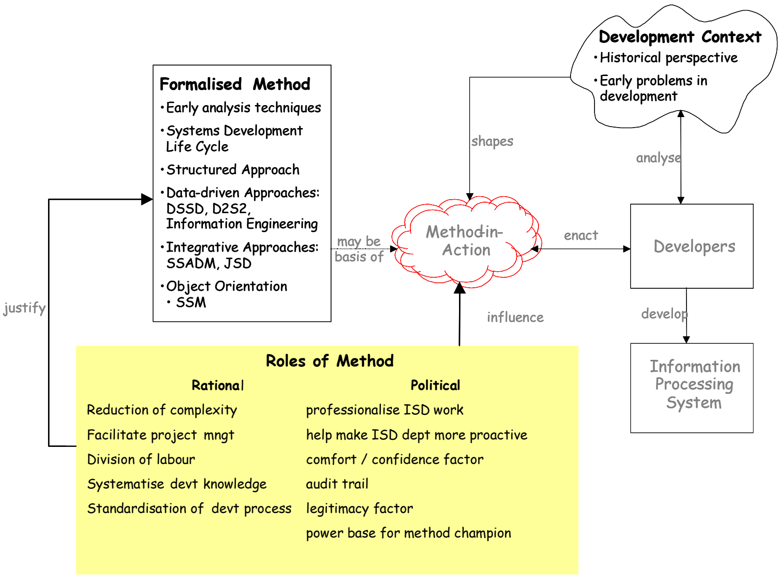
Development Context
• Historical perspective
• Early problems in development
Formalised Method
• Early analysis techniques
• Systems Development Life Cycle
• Structured Approach
• Data-driven Approaches: DSSD, D2S2, Information Engineering
• Integrative Approaches: SSADM, JSD
• Object Orientation
• SSM
shapes
analyse
may be basis of
Method-in-Action
enact
Developers
justify
influence
develop
Information Processing System
Roles of Method
Rational
Political
Reduction of complexity
Facilitate project mngt
Division of labour
Systematise devt knowledge
Standardisation of devt process
professionalise ISD work
help make ISD dept more proactive
comfort / confidence factor
audit trail
legitimacy factor
power base for method champion

Figure 1. A Framework for ISD Method Use (c.f. Fitzgerald *et al*, 2002:90)

Developers is a term used in a broad sense to cover the multiplicity of stakeholders, systems users, analysts, designers, programmers, clients and problem owners. The developer is accorded a central role in the framework, thus reflecting the fact that it is people, not methods, who develop systems. According to the framework, the developer analyses the development context and uniquely enacts the method-in-action to develop an information processing system.

Information Processing System: just as the development context is not always the same, the information processing systems being developed are not all alike. According to Fitzgerald et al (2002:138), a number of families of systems can be identified, and depending on the different characteristics of each family, serves to affect the method-in-action that will be needed to develop them.

In sum, the framework illustrates the nature of systems development, and some significant aspects and factors that need to be considered, including the concept of method-in-action. It needs to be re-stated that the framework is a conceptualisation of the systems development process, making it possible for researchers to reflect on ISD as a complex process influenced by all the framework components and their interactions.

RESEARCH PROCEDURES

This section justifies the use of grounded theory (Glaser & Strauss, 1967) when conducting process oriented research and describes a number of distinguishing features that characterise this approach – inductive, contextual, and processual – that fit with the primarily interpretive rather than positivist orientation of this research.

Reasons for Using Grounded Theory Techniques

Due to the lack of prior empirical research on the topic, a theory building research design was chosen, and sought to illustrate the framework's capacity to illuminate a phenomenon in new or better ways. A grounded theory (GT) approach based on an in-depth case study of one organisation in the financial services industry was considered appropriate given the purpose was to develop an understanding of this under-researched phenomenon. I also chose grounded theory (GT) techniques to analyse my case study interview data because, according to Strauss & Corbin (1998), grounded theorising is well suited to capturing the interpretive experiences of IT practitioners and developing theoretical propositions from them. In the same line of thought, an application of GT is appropriate when the research focus is explanatory, contextual, and process oriented (Eisenhardt, 1989). Similarly, GT has been effectively used in recent IS research (Galal, 2001; Urquhart, 2001; Golicic *et al*, 2002; Douglas, 2003; Hughes, 2004) to develop theory of IS practice.

The Research Design

The first step in the research example involved stating the research questions and identifying a preliminary conceptual model that initially, told me where to look for relevant evidence. The role of loosely specifying a conceptual model prior to the conduct of any data collection is a major point of difference between the research adopted compared to a purely inductive approach. In fact, I chose this approach essentially because it provided a more "tighter" design, in keeping with the following contingencies.

Firstly, the literature provides a good prior acquaintance with the research problem; we know something conceptually about the phenomenon – in this case a bank of concepts providing at least a rudimentary conceptual model of the enactment process. There is a significant body of literature on factors influencing method use and theory on software process innovation, but not enough to house a theory of methodology enactment. I had some idea about how to gather the information and perhaps which questions to ask, and which incidents to attend to closely. As Miles & Huberman (1994:17) state, "not to 'lead' with your conceptual strength can simply be self-defeating". Yet, as these authors point out, no matter how "tight" the design is bounded, focused and organised, qualitative research designs are not copyable patterns or panaceas that eliminate the need for building, revising or 'choreographing' analytic work. According to Miles & Huberman (1994:17) tighter designs are a wise course providing clarity and focus for qualitative researchers worried about diffuseness and overload.

The Case

The study focussed on an Australian bank. The banking and financial services sector was chosen because of the extremely important role that IT plays in the success of companies in this industry, and the bank selected has extensive experience and use of an in-house developed methodology. The selection of the case site was based on a combination of accessibility to the company's IT managers and project members, and interestingness – in the sense that the chosen bank is one of Australia's top 4 banks, and it's IT organisation is considered to be a leading player in providing state-of-the-art IS solutions to customers. The paper uses the pseudonym *The Bank* for the purpose of maintaining anonymity and confidentiality to the case study participants. The unit of analysis for the case study were IT professionals (IT managers and systems developers) within the systems support and systems development division.

Selecting the Participants

I made two major decisions in terms of selecting the participants. First, the population of interest were specified as IT practitioners and project managers. The interviewees were purposely selected based on the researcher's knowledge of the industry, and from discussions with industry figures. The second critical decision was to know when to stop adding participants to the study. One approach for deciding when to stop adding is to conclude the field research when theoretical saturation is reached (Eisenhardt, 1989; Strauss & Corbin, 1998). This approach may be the ideal situation but is difficult for researchers (like grant recipients) who faced the real constraint of time schedules and funding. Hence some researchers must develop alternate approaches to answer the question of when to stop adding interviews. An appropriate alternative stresses the importance of representing the variety found in the population rather than reproducing the proportions of characteristics found. This approach embodies the concepts of *replication logic* (Yin, 2002). As suggested by Smith (2000:78)

> *in research at the organisational level, the researcher includes in the sample a variety of [responses] operating under different conditions to ensure the theory developed is robust and provides explanations of phenomena across a number of different settings. It is the variety of conditions found under replication logic that permit the generation of theory capable of explaining the diversity of situations typically found in organisations.*

Following Glaser and Strauss' (1967) technique of theoretical sampling, three occupational functions within *The Bank* were selected for their similarities as well as their differences. Because the *purpose* of the research was to generate theory applicable to various occupational contexts, differences were sought in organisational type such as those working in development versus those working in a maintenance or support environment. These differences allow useful contrasts to be made during data analysis, which challenge and elaborate the emerging concepts.

Data Collection Instruments

In my role as an outside observer, the semi-structured face-to-face interview was chosen as the primary data source, with observations and documents being minor sources of data. Importantly, the reliance on multiple data collection methods increased the robustness of results through triangulation. The primary goal of the interviews was to elicit the respondent's views and experiences in his or her own terms. As such, the research used the natural setting as the direct source of data and the researcher as the key data-gathering instrument. The data gathering task was to access other people's interpretations, filter them through the researcher's own conceptual apparatus, and then feed a version of the events back to others.

Next, the primary form of data collected – *critical incidents*, and each of the data sources used in this study are discussed. *Critical incidents* are brief descriptions described by individuals about significant or key events that relate to a particular topic. This technique has been used widely in organisational research. Miles & Huberman (1994:115) cite the use of critical incidents and explained that 'sometimes a researcher wants to limit an event listing to those events seen as critical, influential, or decisive in the course of some process'. Several advantages accrued from using this method of collecting data. Respondents were asked to talk about specific situations, events and people. The advantage was that the interviewee could focus and reflect on the described incident posed in the interview. A collection of these responses allowed the researcher to analyse meaningful data that was grounded in the actual experiences, needs and concerns of participants.

Interviews with IT practitioners dealt with the following issues: reasons or motives for using the ISD method; the conditions that shaped their use; and the enactment process itself. The average length of each interview was approximately 1 hour. Interviews were taped (although one interviewee declined to be taped) and were later transcribed by a typist into a word-processed document. Some respondents were interviewed a second time to follow-up on important issues that became evident in the data analysis phase. A total of 30 interviews were conducted with 25 informants from different projects and at varying levels in the organisation. The levels and project types of the respondents are shown in Table 1. 16 interviews were carried out in the first round in December 2003 and a further 14 interviews in the second round in March 2004. Only data from the first round of interviews are used in this paper.

Table 1. Interviews by Title within The Bank, and Project Type

	Job Title			
Project Type or Function	**Project Manager**	**Senior Consultant**	**Consultant**	**Total**
Support / maintenance	2	8	6	16
Development	2	4	3	9
Auxiliary Management	3	2	0	5
	7	14	9	30

DEVELOPMENT OF THE CODING STRATEGY

This section describes how the data was coded and managed. Two procedural issues were of great importance in starting the research: the initial definition of research questions; and the choice regarding *a priori* specification of constructs. Each of these issues is examined in turn.

First, the research questions provided the focus. The aim of the study was directed toward exploring and describing how and why the methods are used, and to develop process theory explaining the enactment process. In pursuit of this aim, three interrelated research questions were initially stated:

> Q1. What issues and conditions influence the decision of IT practitioners to use, modify or tailor ISD methods?
>
> Q2. What are the processes IT practitioners go through when using ISD methods at their work?
>
> Q3. How can these processes of methodology enactment be depicted in a model?

The research questions examined in the literature review provided a guiding focus to the research and permitted the specification of the kind of data to be gathered. This approach conforms with that of Strauss & Corbin (1998:47) who suggest that literature from the field be used not to develop hypotheses, but to stimulate theoretical sensitivity by providing concepts and relationships that are checked out against actual data.

With respect to the second issue of using existing theoretical constructs to guide theory building, a loose conceptual model and its constructs were only used as a starting point. The conceptual model was intended to make sense of the interviews, ensure that possible issues were not overlooked, provide a set of constructs to be investigated, and guide the author's interpretation and focus. However, as stressed by Eisenhardt (1989), although early identification of possible constructs allows them to be explicitly studied in interviews, it is equally important to recognise that the constructs are merely tentative in the theory-building process. In this research, this was found to be true as new issues were identified during data collection that needed to be added to the analysis.

Analysis of Data

The approach to data analysis included two main steps: early steps in data analysis, and within case analysis. The *early steps* in analysis include use of the contact summary form for reviewing the interview, the development of a computerised database using the latest QSR qualitative research software (NVivo v2) specifically designed for storing, indexing, searching and theorising on qualitative data, the arranging and displaying of data in tables, and the development of a coding scheme to organise the data. *Within case* steps involved detailed write-ups for each interview assisted by the identification of critical incidents. Three analytical techniques adopted during each of these two teps are explained below.

The first technique, combining data collection and analysis was the use of the *contact summary form*. This form was useful in revising the case interviews, and once transcribed, reflective remarks were recorded about the main issues identified, and initial thinking around the research questions. These remarks were ways of getting ideas down on paper and of using writing as a way to facilitate reflection and starting the analysis process. Use of the *contact summary form* commenced analysis by presenting an overall evaluation of the case scenario. The initial analysis was performed manually, involving writing up the contact summary sheet with some further development of the coding scheme. The transcripts from each interview were then coded and analysed for emerging themes.

A second set of techniques recommended by Bazeley & Richards (2000) highlighted data management issues, and the application of analytical software to the interpretive analysis of qualitative data. In this research, the process of analysis was assisted by and recorded in a developing database through procedures such as importing rich text files of transcribed interviews, setting up other links in which comments and reflections could be added, coding and sorting the interpretations; and text retrieval of selected instances into the body of the research report.

The third technique was *coding*. While the data is being systematically collected (mainly through interviews), the researcher is confronted with volumes of rich text. A systematic way of handling the words was through coding. In this paper, a code is defined as an abbreviation or symbol applied to a segment of words (e.g. a sentence or paragraph) that captures the description of the phenomenon. Coding, then, is labelling fragments of data by identifying key points, as opposed to coding by micro-analysis of the data, word-by-word, or line-by-line. The selection of key points accords with Miles & Huberman's (1984:57) recommendation as a protection mechanism against data overload, and Glasser's (1992) concern of over-conceptualisation

Coding Strategy

Coding is also a central activity in the process of generating grounded theory. Since publishing their seminal article in 1967, Glaser and Strauss have parted in their views on the application of grounded theory. In terms of specifying the steps to be taken by a researcher in coding and analysing phenomena, Strauss & Corbin (1990) are more prescriptive, opting for a structured set of analytical steps and encourage the researcher to predetermine the general subject of inquiry before entering the site. Glaser (1992), on the other hand, prefers an analytical method that is more general, and regards Strauss & Corbin's (1998) analytical method of forcing rather than allowing emergence of theory. For reasons of a 'tighter design', the research followed Strauss & Corbin's approach, where the author elected in advance to focus the observations and interview questions on a particular issue, such as methodology enactment. In terms of coding process, coding was then oriented around this issue, and a central concept is then sought to represent the nature and dimension of methodology enactment.

As suggested by Miles & Huberman (1994:58) I created a provisional 'start list' of codes prior to fieldwork. Most of the initial coding categories were drawn from the loose conceptual model (the Fitzgerald *et al* framework), the list of questions, and key concepts the researcher brought to the study. To be consistent with the framework, the preliminary descriptive coding scheme developed in this study was divided into five broad categories: roles of method, formalised method, development context, developers & information processing system. The original list was then used to codify and extract the data from the transcript associated with the pilot interview. As a result of this process, I found the need to add new codes. This same format was carried through the entire data collection process (across the 1st 16 interviews) and new open codes were developed for emerging themes.

Two coding techniques need to be discussed. In the first, coding takes place where transcripts are labelled by applying *in vivo* codes; meaning the literal word or phrase used by the respondent becomes the code. However, this research used an alternate approach where a set of pre-defined 'start codes' or researcher-constructed labels were applied to the text that best captured the description of the phenomenon. The development of the coding scheme was an on-going process throughout the transcription of each of the 16 interviews. 35 resulting codes within five major categories emerged from the analysis of the first round of interviews. A portion of the results of the revised coding scheme are presented and explained in Appendix A, displaying the categories, and their coded components.

DISCUSSION, CONCLUSION & FURTHER RESEARCH

One case study of 30 interviews to establish the plausibility of the framework to establish start codes is enough. The findings are not suggesting that we can use the case as a definitive test of the framework. However, the case succeeds in establishing the plausibility of the Fitzgerald *et al* (2002) theoretical framework by providing evidence of its capacity to provide start codes illuminating factors accounting for the use or otherwise of the systems development methodology. The derivation of start codes also provides initial evidence suggesting that the research proceed to the next phase of theory development where the method-in-action process and the derivation of axial and selective codes is specified further into a set of propositions and a graphical model.

This is the first paper in a series of papers describing the process of researching methodology use in a large Australian bank. For the author of this paper, the next phase will involve (i) analysing the remaining 14 interviews, and if the findings further support the plausibility of the coding framework; (ii) the research program will commence further steps of GT by generating selective codes, further categories and identifying a central theme; then (iii) the development of propositions and a substantive theory. In this theory-building phase, the researcher proposes to analyse the remaining set of interviews with the intention of generating a set of propositions explaining the method-in-action process. On the other hand, given the iterative nature of case research, additional analysis may produce conclusions that argue against the framework's plausibility, thereby forcing a new round of theory specification research.

APPENDIX A

The Coding Scheme

Category or theme

Construct explanation

Code

Rational Roles

Rational Factors	Code
Reduction of complexity ISD methods imply a strategy that subdivides the development process (divide & conquer) +	**[R-comp]**
Facilitate project mngt & control .. the influence of business users through their control of the sign-off process +	**[R-proj]**
Division of labour different tasks require different skills, and economies in pay rates depending on skills required +	**[R-div-o--lab]**
Systematise development knowledge capture knowledge from the developer to the methodology so that it can be stored, disseminated, exchanged +	**[R-know]**
Standardisation of devt process .. standardisation of the development process across all projects within a large organisation +	**[R-stand]**
Desirability for ISO certification as part of the quality endorsement process +	**[R-cert]**
Solution to software development … a widely help opinion that methods represent the best way to develop systems +	**[R-best-way]**
Definitional anomalies problems as to what actually constitutes a method -	**[R-def-anom]**
Generalisation without validation an absence of peer review or evaluation of their applicability -	**[R-validity]**
Inadequacies of the rational scientific paradigm …. Over-emphasis on technical rationality at expense of softer, social aspects -	**[R-rational]**
Goal displacement .. . slavish and blind adherence to the method while losing sight of the fact that development of the system is the real objective -	**[R-goal-disp]**
Methods are universally applicable .. failure to recognise the uniqueness of every development situation -	**[R-unique]**
Inadequate recognition of developer-embodied factors .. methods don't cater for critical factors such as individual creativity, intuition, or learning over time -	**[R-individ]**

Political Roles

Political Factors	Code
Professionalise ISD work … insulating developers from conceding to unreasonable deadlines and demands from business user departments.	**[P-prof]**
Make ISD more proactive .. promote the IT department to a more proactive role in strategy formulation	**[P-proact]**
Comfort / confidence .. method use provides for new IT professionals some reassurance that proper practices are being followed	**[P-comfort]**
Audit trail … providing an audit trail of the development process to afford protection if design decisions turn out wrong in the future	**[P-audit]**

Legitimacy factor … use of the methodology enables the organisation to win government contracts, or achieve certification	**[P-legit]**
Power base for method champion .. method champions may use it to raise their profile within the organisation	**[P-power]**

Development Context

Context factors	**Code**
Context & Technology… new technology makes it possible to perform organisational tasks in different ways	**[C-technology]**
Context & Culture .. culture (collection of experiences expressed in a set of beliefs & values). .. a culture will influence what is and what is not possible to do within a specific context	**[C-culture]**
Strategies for change … proactive (opportunity to be exploited) v reactive (responding or reactive to a change situation elsewhere) …	**[C-change]**
Strategies for change … problem solving (getting away from an unwanted situation) v innovation (change is viewed creatively to form a new context)	**[C-change]**
Strategies for change … incremental (on-going daily activity in small steps) v radical (change in large and more focussed steps)	**[C-change]**
Strategies for change … high risk v low risk	**[C-risk]**
Knowing the context … understanding of the context	**[C-knowing]**

Developers

Developer factors	**Code**
Developer awareness of methods …. education and training in method use	**[D-awareness]**
Experience … development experience and prior exposure to, and use of methods	**[D-exp]**
Knowledge of application domain … experience of the banking business domain	**[D-domain]**
Commitment & motivation … Commitment & motivation to the project	**[D-commit]**
Autonomy .. discretion in relation to the development process	**[D-autonomy]**

Information Processing System

System factors	**Code**
System purpose ….	**[S-purpose]**
System complexity .. size, technical dimension, user interaction, etc	**[S-complex]**
System solved or unsolved … common or routine situations v class of system not developed before	**[S-solved]**
Uniqueness or Standard …	**[S-standard]**

REFERENCES

Avison, D., & Fitzgerald, G. (2003) **Information Systems Development: Methodologies, Tools and Techniques**, McGraw Hill, Berkshire.

Bazeley, P., & Richards, L. (2000) **The NVivo Qualitative Project Book**, SAGE Publications.

Beynon-Davies, P., & Williams, M. (2003) The Diffusion of Information Systems Development Methods, **Journal of Strategic Information Systems**, 12, 29-46.

Douglas, D. (2003) Inductive Theory Generation: a grounded approach to theory generation, Electronic **Journal of Business Research methods,** Vol 2(1).

Eisenhardt, K. (1989) **Building Theories from Case Study Research**, Academy of Management Review, Vol 14, No 4, pp. 532-550.

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

Fitzgerald, B. (1998) An Empirical Investigation into the Adoption of Systems Development **Methodologies, Information & Management** 34, 317-328.

Fitzgerald, B., Russo, N., & Stolterman, E. (2002) **Information Systems Development: methods in action,** McGraw Hill, Berkshire.

Fitzgerald, B., Russo, N., & Stolterman, E. (2003) Software Development Method Tailoring in Motorola, **Communications of the ACM**, 46(4), 64-70.

Galal, G. (2001) From Contexts to Constructs: the use of Grounded Theory in Operationalising Contingent Process Models, **European Journal of Information Systems** 10(1), pp 2-14.

Glasser, B. (1992) Basics of Grounded Theory Analysis, California, Sociology Press.

Glaser, B., & Strauss, A. (1967) **The Discovery of Grounded Theory – strategies for qualitative research,** Weiderfield & Nicholson. London.

Golicic, S., Davis ,D,. McCarthy, T., & Mentzer, J. (2002) The Impact of e-Commerce on Supply Chain Relationships, **International Journal of Physical Distribution & Logistics Management**, Vol 32(10), 851-871.

Hansen, B., Kautz, K., & Jacobsen, D. (2003) Information Systems Development Methodologies in Practice, **Proceedings 12th International Conference on ISD Methods & Tools - Theory & Practice,** Melbourne, Australia.

Hughes, J. (2004) Reflections of the Use of Grounded Theory in Interpretive Information Systems Research, **Electronic Journal Information Systems Evaluation**, Vol 6(1).

Iivari, J., Hirscheim, R., & Klein, H. (1998) A Paradigmatic Analysis Contrasting Information Systems Development Approaches and Methodologies, **Information Systems Research**, 9 (2), 1-30.

Iivari, J., & Maansaari, J. (1998) The Usage of System Development Methods: are we stuck to old practices? **Information & Software Technology**, 40 (9), 501-510.

Kiely, G., & Fitzgerald, B. (2003) An Investigation of the Use of Methods within Information Systems Development Projects, paper 34, **IFIP 8.2** Conference, Athens Greece.

Madsen, S., & Kautz, K. (2002) Applying Systems Development Methods in Practice - The RUP example, **Proceedings of the 25th IRIS conference**, August, Bautahøj, Denmark.

Miles, M., & Huberman, A. (1994) Qualitative **Data Analysis: An Expanded Sourcebook**, Sage, Thousand Oaks.

Russo, N., & Stolterman, E. (2000) Exploring the Assumptions Underlying Information Systems Methodologies, **Information Technology & People**, 13(4), 313-327.

Sauer, C., & Lau, C. (1997) Trying to Adopt Systems Development Methodologies – a case based exploration of business users' interests, **Information Systems Journal**, 7, 255-275.

Smith, A. (2000) **Casing the Joint: Case Study Methodology in VET Research at the Organisational Level**, ANZ Journal of Vocational Education Research, Vol 8, No 1, pp. 73-91.

Strauss, A., & Corbin, J. (1998) Basics **of Qualitative Research**, SAGE Publications.

Truex, D., Baskerville, R., & Travis, J. (2000) Amethodical Systems Development: the deferred meaning of systems development methods, **Accounting Management and Information Technologies**, 10, 53-79.

Urquhart, C. (2001) An Encounter with Grounded Theory: Tackling the Practical and Philosophical Issues, in: Eileen Trauth (ed.), **Qualitative Research in Information Systems: Issues and**

Trends, Idea Group Publishing, 104-140.

Vidgen, R., Madsen, S., & Kautz, K. (2004) Mapping the Information Systems Development Process, **Proceedings of the IFIP WG8.6 Working Conference on IT Innovation**, Dublin, Ireland.

Whetten, D. (1989) What Contributes a Theoretical Contribution**?, Academy of Management Review**, Vol 14, No 4, pp 490-495.

Wynekoop, J., & Russo, N. (1993) System Development Methodologies: Unanswered Questions and the Research-Practice Gap, in J. DeGross, R.P. Bostrom, & D. Robey (eds.), **Proceedings of the Fourteenth International Conference on Information Systems**, December 1993, pp. 181-190.

Wynekoop, J., & Russo, N. (1995) Systems Development Methodologies: unanswered questions, **Journal of Information Technology**, 10, 65-73.

Wynekoop, J., & Russo, N. (1997) Studying Systems Development Methodologies: an examination of research methods, **Information Systems Journal**, 7, 47-65.

Yin, R. (2002) **Case Study Research: Design and Methods**, 3rd Ed, Sage, Beverly Hills, California.