POWER AND AUTHORITY OVER SYSTEMS PROFESSIONALS BY THE BUSINESS CLIENT

Bruce H. Rowlands
School of ICT
Griffith University
Brisbane, AUSTRALIA
b.rowlands@griffith.edu.au

Abstract

This paper reports on research into how systems developers enact an information systems development methodology (ISDM) with a focus on describing how enactment is bound up in everyday social and organisational structures. A case study of an IT department within a major financial institution is used to explore the impact of power and authority structures in the enactment of an in-house developed ISDM.The case study develops an argument that authority structures such as policies and work practices embedded within the ISDM are active forces in the systems development process. The findings depict business exercising nearly complete control over the development process and systems developers as playing a reluctant, but submissive role. The study also develops a theoretical framework that integrates elements of a social actor model to provide a more fine-grained analysis concentrating on the relationships among systems developers, the business client, the ISDM, and the environment surrounding its use.

Keywords: Power, ISDM enactment, IT practice, social actor model

Introduction

Power relations between the business client and systems developers are an important topic of research in the field of IS (Markus and Bjorn-Andersen, 1987). This paper addresses one-half of the business-client/systems developer power relation: the power of the business client over the systems developer. In this example, it is illustrated how the business client exercises power over the systems developers by creating organisational structures and routine operating procedures embedded within the information systems development methodology (ISDM) that give them formal authority over developers or foster dependence on them for important resources. Because systems professionals are the enactors of the ISDM and developers of the information systems, it is their understandings, intentions, actions and responsibilities that are the centre of this paper's attention.

Questions about the distribution of power over the development of information systems in organisations have largely been neglected in the literature. Within the field of IS, information and power were considered to be synonymous, and hence those who built business systems were viewed as instrumental in influencing power relationships (Markus and Bjorn-Anderson, 1987). Apart from Saunders and Scamell (1986), Beath and Orlikowski (1994) and more recently Dhillon (2004), the IS research community has rarely challenged or scrutinised this way of distributing authority. In this paper we attempt such a scrutiny by examining in depth, the enactment of a systems development methodology. By examining how one methodology lays out the power relationship between the business client and systems developers, how it distributes resources, how systems developers respond to its prescriptions, we can gain insight into how systems developers enact a methodology in practice.

This paper disagrees with the notion of Markus and Bjorn-Andersen (1987) that developers of IS have power over users. They may have once, but trends in software development and business environment appear to have reduced the business client's dependence on IT for expert services. This research provides grounded evidence from a large organisation in the financial sector (*The Bank* – a pseudonym) with an internal software development division, that there has been a power shift with the business client able to exercise considerable power over systems developers. In *The Bank* it will be illustrated how the mandatory use of a ISDM constrains systems developers through the necessity of gaining sign-off and further funding at each stage of development enabling the business client to exercise control over the development process. In this paper, we will argue that it is the policies and practices embedded in the ISDM that constitute the structural exercise of power. Also, it is argued that the constraints based around the accepted and everyday use of a methodology by systems developers obviates the need for more overt or direct forms of control.

In this paper we define power in terms of behavioural outcomes, rather than the intentions or perceived legitimacy of behaviours involving power use. From our perspective, to say that the business client has exercised power over systems developers means that the developers behave differently from the way they would have if not for the business client (Sillince and Mouakket, 1997). We also define systems professionals within *The Bank* to include all those individuals and groups, both inside and outside *The Bank*, who 'consult' with the business client; that is, assess their needs, propose solutions, and develop software systems. This broad category of systems developers includes in this case study project leaders, systems analysts, analyst/programmers, IT managers, and the methodology support personnel. Our definition of the business client includes all those user departments who operate and interact with the software systems to achieve organisational goals. The business client generally initiates the development or enhancement of systems but most importantly, funds the development of these systems.

The case study reported in this paper builds on research suggesting that ISDMs significantly inform and shape the cognitions and actions of organisational members engaged in systems development (Robey and Markus 1984, Markus and Bjorn-Anderson 1987, Hirschheim and Klein 1989). The case develops an argument that institutional forces embedded within development methodologies are active forces in the systems development process. The influence of methodologies occurs both through the material constraints and prescriptions of process mandated by the methodology, and through the experiences and learning from previous use of the methodology that shapes developers' approaches to using the ISDM in their workplace. The case builds on the empirical findings of many studies of technology appropriation that how technology is appropriated is always problematic in light of pre-existing work practices, work culture, institutional forces and other socio-technical elements associated with the technology (*c.f.* Gosain, 2004). We view ISDMs as social institutions that exert their own types of agency that interact with human agency in the systems development process. The case analysis sheds light on the emergent

aspects of the pre-existing ISDM and the interaction between disciplinary and human agency in IS development (Chae and Poole, 2005).

Information Systems development methodologies (ISDMs) are reported to play a useful role in guiding the development and on-going maintenance of software. However, central to the philosophy of the ISDM is the assertion that systems development is founded upon dialogue and agreements which, it is argued, provide the best guarantee of predictable patterns of action that then can be successfully modelled and replaced by information systems (Elkjaer et al, 1991). Likewise, the value of established methodologies for systems development have been challenged by approaches which question the adequacy of their assumptions and the efficiency of their prescriptions (Truex et al, 2000). Yet, there appears to be few practice demonstrations or case studies illustrating the method enactment process in their social and organisational contexts. Researchers have long called for research on methodologies in real life organisational situations (Wynekoop and Russo, 1997; Iivari and Maansaari, 1998; Beynon-Davies and Williams, 2003) and they continue to do so (Kautz, Madsen and Nørbjerg, 2007). With this call in mind, this paper addresses two related research questions: (1) how do institutional processes and social structures shape the method enactment process; and (2) how does the ISDM reinforce the primacy of the business client?

To answer such questions requires investigating the conditions and rationales in the institutional context that reinforces the primacy of the business client. We need to examine the authority and power afforded the business client. We need to inquire into the circumstances within which systems developers enact an ISDM, and examine the conditions that leave systems developers with little control over the development process and technological resources they use in their work. Inquiry into these institutional elements are critical areas of research for the field (Beath and Orlikowski, 1994; Silva, 2007).

Insight with respect to these questions will be derived from a case study of the enactment of an in-house developed methodology in a large IT department of a major Australian bank. The study of systems development and power discussed in this paper was conducted in an organisation that implemented and operated an architecture of centralised mainframes, primarily custom software development in-house, and the use of a mandated ISDM that required the services of professional systems developers. The research approach makes use of Lamb and Kling's (2003) *social actor model* by illuminating an aspect of information systems practice – methodology enactment – at four levels: individual, project, organisational, and industry. This novel theoretical approach when combined with in-depth empirical research extends our current understanding of contemporary ISDM enactment. The paper also operationalises the *social actor model* as first demonstrated in Rowlands (2006) in terms of mapping the case interview text to the model's sixteen constructs. These are important findings as the *social actor model* is relatively new.

This research is also important for the following three reasons. Firstly, ISDMs are purported to be valuable and that according to Avison and Fitzgerald (2006) ISDMs address many of the functions required in developing information systems such as determining or regulating activities and providing resources to support actions. We also believe that the significance of ISDMs resides in its expanding role in the regulation of human affairs. ISDMs as artefacts of the systems development process articulate and facilitate the reproduction of relations that are institutional, political, and socio-economic (Hirschheim and Klein, 1989). Second, previous research on ISDMs tend to focus on the features of the methodology and systems developer's behaviours while underemphasising the role of context and institutional structures (Chae and Poole, 2005). Among the many studies of ISDM enactment, few pay attention to the role of context, social and organisational structures embedded in the systems development methodology. Orlikowski and Iacono (2001) argue that most IS literature treats IS as separable from the social and organisational contexts in which they are instantiated, appropriated, and enacted. When ISDMs are viewed as technical artefacts that are relatively stable, discrete and fixed, it is easy to view the enactment of an ISDM as an independent event. This view encourages the belief that institutional structures have little to do with the enactment of an ISDM in the workplace. Technocentric views of the systems development process omits the role of human agents and the existing organisational and social contexts in the enactment of systems development artefacts. The third reason is that the broader IS literature is dominated by individualistic theories of human behaviour (e.g. TAM) (Sawyer and Chen, 2002). In contrast, this case adopted an analytical perspective[1] (Kling et al. 2005:7) in that the intent was to contribute towards developing socio-organisational theory about ISDMs in institutional and cultural contexts. The

---

1 An analytical perspective is one that leads to developing theories about ICTs in institutional and cultural contexts, or doing empirical studies that are organised to contribute to such theorising. In contrast, a normative perspective to research is one whose aim is to recommend alternatives for professionals who design, use, or make policy about ICTs.

approach to theory building, with an emphasis on concept development and operationalisation made use of *social actor model* to illuminate an under-researched aspect of information systems practice – how ISDMs reinforce the primacy of the business client – within a large Australian bank. Based on the findings of a detailed case analysis, this paper details a fresh approach for understanding methodology enactment across multiple levels by conceptualizing the systems developer as a social actor who engages in multiple uses of the ISDM in an organisational context. By taking a social informatics[2] perspective, the paper provides a grounded description of method enactment showing how social context and the power relations structurally embedded within the ISDM influenced enactment in a large IT department of a major Australian bank. The research provides empirical support for modelling systems developers not as users of a ISDM, but as social actors who are engaging uses of the methodology, while embedded within an enabling and constraining social context with individual agency to help shape that context.

The rest of the paper is organised as follows. In the next section we review the literature on the enactment of ISDMs in practice, and in particular, studies of how ISDMs distribute power. In the following section we discuss our theoretical approach to the inquiry and present an argument that ISDMs are social institutions that exert their own form of agency in the systems development process. We then describe the research approach. The case findings are then presented and analysed. In the final section we provide a closing discussion and conclusion.

Review of Literature

Methodology Enactment

A small, but growing body of research has been conducted on the specific topic of methodology enactment. The work that has been carried out is limited in its ability to consider the complex social and organisational context of methodology enactment. Most field research on the use of methodologies has neglected the messy and complex way people work and live, and the dynamics by which power and authority processes and social structures shape ISDM enactment.

In a major publication in the field, Fitzgerald *et al.* (2002:12) proposed a framework to understand the complex nature of systems development and how methods are enacted in practice. Their framework is comprised of six sets of issues, and these authors stress the importance of understanding the myriad of factors and influences relevant to the systems development context and the effect they have on the development outcome. The pivotal component of their model, the *method-in-action* process, representing the enactment of the formalised method is not described, and cannot be described in a general model as each project, set of methods used, and the dynamics of the organisation are particular and unique.

In a study of how a widespread object-oriented ISDM (RUP - the rational unified process) was adapted and introduced in an organisation, Backlund (2004) identified interactions and their relationships to work practices of an actual project as the interplay between implicit and explicit knowledge. Backlund (2004) took the perspective that systems development is best conceptualised as knowledge work as proposed by Iivari (2000) that the role of the ISDM is a means for managing knowledge about the systems development process. There is also a growing argument that IT practice should be seen as more than a technical activity and as argued by Goulielmos (2004), method enactment can and should be understood as a complex social activity that is influenced by the organisational and institutional context in which it takes place. As noted in Aydin *et al* (2005) context played an important role in their study of the adaptation of an agile information systems development method — dynamic systems development method (DSDM). In a further contribution to the discussion about the enactment of ISDMs, Madsen et al (2006) focussed on the unfolding development process, activities, and method elements that comprise this process. They describe the 'emergent method' as a process of social moderation of methodology use, covering both the change of methodology as formally prescribed through use, and the change of its users' actions that result from their interactions with the methodology. Madsen *et al* further conceptualised methodology enactment as a process of organisational innovation. Madsen *et al's* (2006) conceptualisation portrayed the role and usefulness of methodologies as a means for communication, coordination and (re)direction, rather than as a rigorous and rigid means for control. This paper continues the discussion about the enactment of ISDMs with an emphasis on how

---

2 According to Kling at al (2005), Social informatics (SI) is a trans-discipline – spanning fields such as management, sociology, and information systems. SI is neither a theory nor a method. SI is an approach to understanding, theorising and engaging ICTs.

how systems developers' concerns and every day work practices are shaped by institutional structures including the ISDM itself and the power relations existing between the business client and systems developers.

Power

Jasperson *et al* (2002) provide a definition of power based of the work of Hall (1999). In general power has to do with relationships between two or more actors in which the behaviour of one is affected by the behaviour of the other. Recognising that power has multiple meanings Jasperson et al (2002) defined power to include authority, decision rights, and influence. These authors also identified a number of paradigms underlying power research and a number of lenses to better understand researcher's views regarding the causal structure between IT and organisational power. This research adopts a pluralist perspective (Jasperson et al 2002) of power, where actors are assumed to have different, potentially conflicting interests. This perspective defines power in terms of actors' ability to influence others' behaviours. For instance, a pluralist conceptualisation of power assumes that resources, possessions of resources, and the resulting dependency relationship are characteristics of an objective reality (i.e. features of the social context are identifiable and apparent to all reasonable participants). As will be reported in this study, system developers felt that they derived what power they exercised from their knowledge of and access to technology. In contrast, the business client used their control of financial resources via the sign off process to guide systems development, and to maintain authority over this process.

In this paper, we consider two types of actors: systems developers and the business client. While the motivations of both types of actors vary among individuals and the situation, we contend that developers and business clients rarely have identical or similar interests. For instance, systems developers are mostly computer professionals – who have received formal training in IT, work in a separate specialised section within the organisation (or may be employed by an external software house), and see themselves as technical experts turning requirements and objectives into a constructed product, the system. On the other hand, the business client is motivated by profit, attaining organisational objectives, and the procurement of software systems at a market price.

Only a few studies focus on the power relations between the business client and systems developers in their social and organisational contexts but Robey and Markus (1984) and Markus and Bjorn-Andersen (1987) provide some reference of its occurrence. Markus and Bjorn-Anderson (1987) provide a framework to identify different forms of exercised power to make business clients and systems developers more aware of the influence of power. Their study focussed on role-based power and the rational exercise of improving interactions and fostering understanding between superior and subordinate. Even fewer studies have directly addressed the enactment of methodologies in these contexts. Nandhakumar and Avison (1999) provide one example highlighting various social issues and influences relating to organisational context such as developers knowledge about methodologies, implicit social norms, and organisational form and culture.

Theoretical Framework

Social Informatics

Our attempt to understand the enactment of an in-house developed methodology in a large-scale organisation made us aware of a need for theory and analytical framework that addresses issues of the technological artefact, the role that actors play in enacting the technology, and at different levels of analysis. Social informatics research has devoted attention to how information technologies enter into action. One of the most prominent lines of work is research on conceptualising the user as a social actor (Lamb and Kling, 2003; Lamb, 2006). Drawing on the work of Scott (1995), Lamb (2006) describes how the social actor concept has been theoretically supported by institutionalist approaches, whereby institutions provide a framing context within which social actors make constrained choices about ICT use, particularly when they are situated in organisations. According to *social actor model*, people's individual autonomy (their agency) and their behaviours are shaped by the social norms, institutional forces (routine work practices), and the social and physical structures that surround them. This approach reflects what Orlikowski and Iacono (2001) identify as the 'ensemble view' of technology where technologies are components of a more complex socio-technical ensemble that include people, work practices, and institutional and cultural factors. In terms of this research example in *The Bank*, structure includes work procedures mandated by the ISDM, the day-to-day interactions within and among project groups, and authority structures based on power and expertise. Given this

view, systems developers can be seen as complex social actors acting in constrained ways, rather than simple "users" of the ISDM (Lamb and Kling, 2003), and where the ISDM operates largely as an institution or structure around which systems developers operate.

Two Key Assumptions of Social Informatics

A fundamental assumption of a social informatics (SI) perspective underpinning this paper is that ISDMs are more than just technical artefacts but are social institutions that exert their own type of agency, and that ISDMs also interact with human agency in the systems development process. The starting point of our discussion is a view that enactment of an ISDM is an interplay between two types of agency: human agency and disciplinary agency.

Pickering's (1995) theory of practice of science (cited in Chae and Poole, 2005) provide key insights for our analysis of agency in systems development. Pickering argues that agency refers to a thing or person that acts to produce a particular result. That is, agency at its base, is the ability to do something or have effects. He distinguishes three different types of agency: the material agency of the natural world, which acts via natural laws; human agency, characterised by human intent, reflexive monitoring of action, and meaningful construction of the social world; and disciplinary agency, in which the agency of a discipline – such as systems development – leads people through a series of actions and also neutralises these actions for them.

Disciplinary agency is defined as the shaping and channeling of human action by conceptual and cultural systems. Disciplines are bodies of knowledge that preserve concepts, practices, and values that can be employed in action (Chae and Pole, 2005: 23). Systems developers therefore work within disciplines that provide scaffolding for their actions. Disciplines provide generalisable procedures applied in the production and reproduction of artefacts, and are largely based on social structures. As this case will demonstrate, the disciplinary elements used in systems development are neither fixed or predetermined, but always emerge in action. These institutional structures of the ISDM do not just impose constraints on development process; they are also enabling to the extent that they provide a repertoire of already existing institutional principles of work (e.g. conventions, practices, common understandings, power relationships) that human agents enroll in their activities. As Orlikowski and Robey (1991:159) put it 'systems developers draw on the values and conventions of their organisation, occupation, and training to build information systems…. They are informed by information systems development methodologies and knowledge about their organisation to build information systems.' In deciphering the role of agency in our case, we argue that the local methodology presents a unique combination of human and disciplinary agency.

A second fundamental assumption of a social informatics (SI) perspective underpinning this paper is that ISDMs and people can only be examined fully through socio-technical and institutional perspectives, and that the technical components of the methodology cannot be fully understood separately from the social and organisational context in which they exist. According to a social informatics perspective therefore, we define enactment of the ISDM as a process in which social actors (systems developers) respond in a dynamic interplay between social context and their intentions to 'determine' a systems development approach for a specific project situation. This interplay is situated in a social and historical context and is bounded by physical surroundings and technological artefacts including the ISDM. Explicit objects produced by systems developers during the methodology enactment include a new software system, an enhanced or maintained system, system documentation, orally expressed ideas, and in some instances, a newer version of the methodology itself. Given this definition, the case analysis applied the user as *social actor model* as a conceptual lens to understand methodology enactment within a single organisational setting. The primary strength of this model is that it is context-centred, and conflates people's interactions, their information environments, and their technologies as the basic unit of research analysis.

Social Actor Model

In terms of our case study, social actors are professional individuals (developers holding university qualifications in IT) performing a role (analysts programmers or project leaders), groups of firm members acting in concert (a project team or an IT department), or organisations (*The Bank*) interacting with industry regulators or industry standards (Basel and Sarbannes-Oxley). The *social actor model* describes four dimensions that connect actors to networks and environments and position their enactment of ICTs as integral to their organisational interactions. What follows is an application of the four dimensions to the specific characteristics of the case:

Affiliations represent inter and intra-organisational relationships created and supported by organisational members as a result of their day-to-day activities as part of the organisation. Systems developers work together with the methodology comprising social networks. These networks exist within *The Bank* but also apply to the IT and financial industries as well, and to a wider national and international context.

Environments. The environment an organisation operates in is formed by the kind of affiliations it has formed with industry, financial institutions and its clients. Methodology enactment needs to recognise the regulated and/or institutionalised practices of *The Bank*, and other associations that circumscribe organisational action.

Interactions. Systems developers see themselves as organisational members working with others (clients, and business partners exchanging information) enacting a methodology (and other media such as email, telephone, web sites) in support of their interactions. Information and resources are mobilised as systems developers engage with *affiliated* organizations.

Identities. Systems developers regularly enact ISDMs to compile and present information to various affiliates. In so doing, they create an identity for their organisation and for themselves. Systems developers are therefore defined by their avowed presentations of the self and ascribed profiles of organisation members as individuals (analyst programmer) or a collective entity (IT professional).

Each of the four dimensions of the *social actor model* are further comprised of four behaviours. These sixteen behaviours were used to develop the initial coding scheme for the qualitative analysis of data. The *social actor model* helped make sense of what occurred in the field, provided a set of constructs to be investigated, and guided the interpretation and focus.

Research Approach

The research approach adopted in this study is that of an interpretive case study (Walsham, 1995). As pointed out by Kling *et al* (2005), people's interpretations of an ICT are based on prior beliefs, and the perceived new opportunities and demands it creates. How people interpret an ISDM is important because systems developers with different interpretations will enact the ISDM differently. Therefore, an interpretive case study was chosen to produce a subjective albeit shared (between the researcher and the interviewee) understanding of phenomena. Generalisations from the setting to a population were not sought; rather, the intent was to understand the deeper structure of a phenomenon, which it is believed can then be used to inform other settings.

The research study was carried out in a large 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 in practice of an in-house developed information systems development methodology (ISDM). Importantly, the banking industry is highly technical, highly competitive, highly regulated and institutionalised. 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 four banks, and its IT organisation is considered to be a leading player in providing state-of-the-art IS solutions to customers.

The sampling strategy for the interviews included a combination of purposeful and theoretical sampling (Schwandt, 2001:232). Three occupational functions within *The Bank* were selected for their similarities as well as their differences. The unit of analysis for the case study were IT professionals comprised of project managers, senior consultants, and consultants within the systems support, new development, and method support divisions. To enhance the credibility[3] of the interview process, all semi-structured interviews were guided by an interview protocol and interviews with systems developers dealt with the following issues: reasons or motives for using the methodology; the conditions that shaped their use; and the nature of their relationships with persons whom they work with across the organisation. The average length of each interview was approximately one hour. Interviews were taped. A total of thirty interviews were conducted with twenty-five informants from different projects and at varying levels within the organisation. A further tactic to ensure credibility was to submit the interpretations to the scrutiny of the individuals upon whom they are based, and to seek their responses to its authenticity – known as member checking (Schwandt, 2001:155). In the majority of cases, each face to face interview was complemented by

---

[3] Credibility is one criterion judging the quality or goodness of qualitative inquiry (Schwandt, 2001:258)

a follow-up email to clarify issues and to obtain supplementary information. The levels and project types of the respondents are shown in Table 1.

To commence analysis, the researcher created and operationalised a list of codes prior to fieldwork. By operationalisation, we mean that the *social actor model* was expressed in coded elements that could be identified in textual analysis. Sixteen codes were drawn from the *social actor model*. 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. In attempting to gain theoretical understanding of the complexity of the text, the *social actor model* was used in a form of content analysis (Schwandt, 2001:34) where the text was systematically listed, coded and categorised according to the sixteen behaviours of the connected and situated individual. The list of researcher-constructed labels that best captured the description of the phenomenon was then applied to the text to codify and extract the data associated with each interview. This same format was carried through the entire thirty interviews. This approach to analysis accords with Langley's (1999) quantification strategy where descriptive patterns of events could be discerned, counted and the *a priori* model mapped. According to Miles and Huberman, (1994), the counting tactic is relevant in qualitative content analysis to verify hunches, and to keep the researcher analytically honest, however space limitations limit the inclusion of supporting numerical evidence. In terms of data management issues, the process of analysis was assisted by and recorded in a database through procedures such as importing chunks of transcribed interviews, adding comments and reflections, sorting the interpretations by code; and text retrieval of selected instances into the body of the research report. Following data collection and initial analysis, the author developed and shared a case report (including a case summary and preliminary elements of analysis) with a key and current project manager involved in the use of the ISDM. This manager commented on the report and gave confirmation of many points and qualifications of others. Drawing on this report, the interviews, the scrutiny of informants, and the researcher's relativistic and subjective understanding of the case phenomena, the paper presents the analytical case.

Table 1. Number of Interviews by Job Role within *The Bank*, and Project Type.

Occupational Type or Function	Job Title			
	Project Manager	Senior Consultant	Consultant	Total
System Support	2	8	6	16
New Development	2	4	3	9
Method Support	3	2	0	5
	7	14	9	30

Case Analysis and Findings

Institutional Context

The Bank is large in terms of Australian corporation size and is old, traditionally stable and bureaucratic. Within *The Bank*, the particular site selected for study is their large-scale IT division carrying out new development and systems support. *The Bank's* IT division consists of approximately 700 people half of whom work in application support. In projects the development expertise is very often within the support team. Each development and support team has a project manager who reports to a business unit department manager who has overall control of the project through budget and a stage-gate funding approval process. The size and composition of project teams usually consists of core people on the project e.g. project manager, business analyst, solution designer, developers and sub-providers; and the other teams that interface with this application: business clients, hardware vendors, the telecommunications provider, and other partners who may be involved in outsourced business processes.

The types of systems built within *The Bank* range from consumer banking (internet banking, personal loans, credit cards) to corporate banking (trading in the futures exchange, commodities, currencies, trade finance, etc), investment banking (shares, funds, fund management) and internal accounting (general ledger, payroll, etc). A trend is toward implementing purchased software packages involving minimal customisation instead of building in-house bespoke systems. Many systems have evolved over the last 30 years and are built around the core banking applications that

keep track of and process customer accounts. Most of these systems still run on the legacy IBM mainframes running MVS, CICS, IMS, and DB2. Because of the size and diversity of the banking applications, the duration of enhancement and new development projects can range from few weeks up to 3-4 years and costs range accordingly up to the many millions of dollars. In addition to the acquisition or development of new systems, existing systems are enhanced to work with and support new applications.

The ISDM

To develop, customise or maintain these systems, the IT division has developed and documented an internal methodology applicable for all development and maintenance tasks. The methodology, known as *The ISDM* ― a pseudonym, is based on traditional 'waterfall' lifecycle phases. Maintenance is not seen as a phase of the lifecycle, but an iteration of software evolution. The methodology is also aligned to an in-house project management methodology. New versions of the in-house methodology are being updated and introduced on a regular basis as development approaches evolve. The use in practice of the in-house methodology is mandatory and covers all new development, package acquisitions and any planned changes to existing systems, except urgent fixes. An Intranet site including the manual and templates of all the documents required at different stages of the lifecycle is made available to all systems development staff.

The methodology itself does not mandate particular tools or techniques. The scope of the ISDM covers the entire systems lifecycle for many different types of systems, clients, and environments with the major outputs being documentation and fully implemented systems. An emphasis within each phase and sub-phase of the life-cycle is on producing documentation.

Two key features of ISDMs are the development lifecycle and techniques for monitoring the transition from one phase to the next within the lifecycle (Avison and Fitzgerald, 2006). Lifecycles logically define the phases involved in development while the ISDM is intended to ensure the translation of system objectives into operational systems within constraints of schedule and budget. This ISDM mandates that systems developers complete technical specifications and deliverables to ensure approval and that resources are allocated before commencing on the next phase of development. Specifically, the transition from one phase to the next requires a formal "signoff" by the business client when work on one phase is completed.

The Case

The user as social actor model involves four dimensions ― interactions, identities, affiliations, and environment that characterise organisational members and their ICT enactment context. According to Lamb (2006) *interactions* and *identities* relate organisationally situated individuals to others and to the ICTs they use to interact with and present themselves to others. The second two dimensions ― *affiliations* and *environments* relate people to their organisation, and to the industries and environments of those organisations.

We use this multi-dimensional conceptualisation of the social actor to guide our qualitative analysis of data. The case illustrates the constraining and enabling aspects of the ISDM and the organisation in context and through empirically supported examples we illustrate how each dimension manifests itself in the day-to-day actions of methodology enactment within *The Bank*.

Interactions

According to the *social actor model* individuals are involved in networks that take shape within and among organisations. Networking refers to the *interactions* where organisational members work and interact with others (affiliates) using a methodology (and other media) in support of their interactions. In terms of *The Bank*, the ISDM is seen as a vehicle to bring together project members and coordinate their tasks when interacting with clients, industry bodies and business partners. The *interaction* dimension was by far the most prominent in the interviews as it describes systems developers in their day-to-day work role networking and relating themselves to others and to the ISDM they use to interact with. The analysis identified a range of interaction behaviours such as producing documentation, communicating, and acting in constrained ways.

According to *social actor model* organisational members seek to communicate in legitimate ways. In its most visible form, the ISDM mandates documentation [IN-DOCN] throughout all phases of the development lifecycle and

requires project leaders to call for meetings, both formal and informal among affiliates to review project status and sign-off on documentation. The methodology in this sense is seen to facilitate project coordination and is a mechanism to review progress, as expressed by a project leader:

when it comes to meetings with various participants they know that what they'll get from the meeting will be documented and they know that the document is a part of method deliverables [2.4]

In terms of producing documentation a repeatedly cited comment was that the methodology required too much attention to paper work. While the aspect of providing a paper-based audit trail was appreciated, many commented that the amount of documentation [IN-DOCN] required was excessive and represented negative reasons for not wanting to comply with the methodology. One development programmer commenting on the technical level of design specifications required and the amount of effort required, felt that documentation [IN-DOCN] often overwhelmed the business client, but agreed that for developers, detail was necessary:

Let's say you need to have a few alterations. You need to send them out to review, to get them signed off.... in some times the business didn't even look at it because they think we produce it in more technical jargon than they understand. So they just sign off on the front page saying 'that's fine'. But for our technical purposes, we go into detail, we need to produce it in detail to send to other departments.[3.3]

While there were many more mentioned down-sides to producing the documentation than positives, many commented that the technical details captured can help later in the project when there is a need to refer back to the documentation. For instance, a support programmer saw valid reasons for documentation [IN-DOCN] being mandated and in seeking approval at all stages of making production changes, but he too saw the ISDM as a form of control:

we do programme changes, we do lodgements, we have to do checklists. The point of doing checklists is for control as to what we're lodging. So you just do your programme changes and then you lodge it, but if you forget about it later on and there's no official record people get confused. The changes then are hard to track down. So we've got these types of controls to make things easier. .[5.2]

The same programmer who commented on the methodology as a control mechanism also saw the amount of documentation required as a lot of red tape and involving excessive time:

basically the main argument against this red tape or what is seen as a red tape is the time factor involved. ...obviously the more approvals you need the more people you need to contact. If certain people aren't around then it takes longer for the approval to come through and if that goes back to the business, they complain about time and you have to explain to them about the time it takes to get these approvals...[5.2]

However, the main reason cited for seeking to communicate in legitimate ways was to get documents [IN-DOCN] signed and gain approval to commence the next stage of development. To do so, there needed to be visibility of development work as a project leader commented:

... producing a document is one way of providing visibility of what's actually happening and with all the formal documents that are required to be signed it's sort of approval to go to the next stage and that the work can be done. Also with funding. Project funding is dependent on these documents having to be produced. In a large scale development the funding is very important in each phase. So therefore you have to produce some kind of deliverable to prove what you've done.[9.1]

A further prominent example where organisational members sought to communicate [IN-COMM] in legitimate ways was the use of common terms (based on deliverables) that the ISDM provided. The interviews indicated that while the methodology provided organisational members with a checklist of deliverables that had to be produced in different stages of the project, the ISDM provided a terminology, so when developers talked among themselves and with the business client people knew the language of the business. Many developers spoke positively in terms of how the methodology facilitated communication [IN-COMM] within *The Bank*. The following excerpt provides a concrete example of how documentation was used to share ideas between a development group and the business client. The documents as a communication tool [IN-COMM] facilitated the process of requirements gathering and project costing. In other words, the methodology provides a common language and is consistent across the organisation.

Business generally are very vague about systems – they know what they want but they don't exactly know what is involved or know if it's feasible. Because they don't know the internals like how the programmes are going to work

or any limitations and those kind of things… they're not that clear with their requirements. That's OK, so we want to communicate our work in a certain way. So we take it from there and understand what they really need. We document the implications. Then they [business] check our documents, and go "OK. This is going to take this long, and that is going to take that long". This is so complex…. but it gives the business the option to review what they're asking for and also so they can see the costs, and what is involved, how big the task is. Then they come back with what they want us to do.[3.4]

A further dimension of the *social actor model* is that while organisations are connected as networks, within those networks, relationships with other actors are often in misalignment or even in conflict. This condition puts pressure on organisation members to develop interactions that presents information in an acceptable, standardised format. *Social actor model* asserts that when this information is exchanged in this format it is made actionable. In terms of this case study, this can be translated as the methodology mandating the generation of specifications that become actionable documents [IN-ACTION] requiring a sign-off at each stage. IT developers viewed sign-off in two ways: positively – to gain approval so that work could commence on the next stage; and negatively – it's a way whereby business clients maintain power and control over the development process. Producing documents, getting approval and sign-off to commence the next stage was seen by many in IT as a major functional component of the methodology, and a major reason for having a formal software development process, as one project manager described:

before you go to the next phase you have to go through the proper approval process again. Usually its a form – you know, fill out this form. You're to say how much money you've spent so far… should we proceed? What is the estimated cost? So these are the kind of steps we have to do at the end of each phase. And without getting a formal approval I couldn't proceed. [3.2]

However, another project manager mentioned another functional role of the methodology, through sign-off, was for the business client to keep financial control of the project:

…you have to get sign off at various points. Yes, the methodology is used by the technology people to build things. But, before you can get funding for the next stage the technology group needs to provide to business things for the project to then proceed to the next phase. So if you want funding to go on further, you'll need to do things. So it [the ISDM] forces you to do things. [7.4]

However, many in IT commented that the process of gaining sign-off was an impediment to delivering systems on time, as the same project manager commented:

the areas where things can be a real barrier is the sign-off process. So let's say we produce a design document and we send it out for review and sign off. Well, you might then have to wait a week or two weeks for other areas to review it and sign off and chase them up because they've all got busy lives as well. So you're in a constant spin trying to get things resolved and issues being raised and closed off. You reissue the document, so you could spend a month chasing all that stuff through.[7.4]

The interviews also identified some difficulty in incorporating a new agile development approach into the methodology especially for web-based applications. Because of the differences in the new techniques, to modify the method to suit local contingencies of the development approach involved intensive personal contact with the method support department. Especially with new development teams there was the feeling that the existing ISDM being obsolete was due for an overhaul. Asked what would be required to introduce an entirely new methodology with the existing method done away and phased out, the method support manager replied:

.. it'd be a big effort [8.1]

However, to introduce a new ISDM would involve serious change management initiatives and this would also involve gaining business agreement who enact the current methodology as a form of control. They would have to be convinced as the method support manager commented:

This is a business and the methodology is part of a control mechanism. .. we need to show business that there are better things out there, if there are, and will enable them to not only get the controls that they do currently have but be more productive and more appropriate for the way that they do their work.[8.1]

According to the *social actor model*, organisational members perform socially embedded, highly specialised actions on behalf of the organisation. These exchanges and interactions take a form of stabilised work practices. Embedded within the ISDM were a set of routine patterns of work. Another prominent interaction code among the transcripts

was that using the ISDM and following these routine patterns of work dictated and constrained [IN-CONSTR] their role within *The Bank* – it told developers what they must do – it's *The Bank*'s way thereby enforcing the culture, rules, regulations of *The Bank*, and the authority structure of systems development, etc. For example, one senior analyst believed that business used the methodology as a form of control by locking development teams into unreasonable schedules making it very difficult on the developers and in the end, producing poor quality systems because:

.. it means that your development time is locked in.Usually, your technical needs are bolted. You're told that you've got to develop a system, so there always has to be an estimate. So what it means to me is that once you've signed-off on a certain amount of money those requirements will be delivered by the end of the project. We're sort of locked in. [6.1]

And, the upshot of this work practice, according to the same analyst was that in a typical project:

we've got this time frame, we have to deliver this project. What you typically do is try and develop 100% of it but what you end up doing is developing 100% of it at 80% quality, instead of developing 80% of it at 100% quality. So you have this thrashing period in the last months of the project where developers are working long hours and making mistakes, causing errors which cause more problems, and so it goes on.[6.1]

The above situation provides an example of a conflict of interest in which the business client achieves their objectives (the introduction of a business system on time) to the relative disadvantage of developers (unreasonable working practices and poor quality code).

Affiliations

According to Lamb (2006) networks are a basic configuration for organising social, economic and political exchanges. The *affiliations* dimension drew attention to the interaction practices migrating within *The Bank* and across organisations. On a network level, as relationships change and new social actors are enrolled, interaction practices migrate within and across organisations. For example IT staff regularly work with external organisations when aspects of projects have been outsourced; or when dealing with contractors brought in on a needs basis. Developers also liaise within *The Bank* with business clients. These interactions bring about change [A-CHANGE], and through this change it shapes how the methodology was used, and in particular how it evolved over time.

Interaction practices cause conflict when methods are migrated within and across *The Bank* with *affiliations*. According to Lamb (2006), because relationships are multi-level, multi-network (i.e. group, inter-group, organisation, culture), and as social actors in their various forms and functions become enrolled in multiple networks, they begin to translate their interests. Within *The Bank*, the *interactions* between the social actors and their differing interests is demonstrated in the following excerpt where the method support manager is commenting on the power [A-POWER] relationship between business and IT:

Business sometimes do hold development to ransom, so to speak. So that's another part of the culture. Really they should be working together to try and deliver solutions rather than using contracts as ransom to force them to do something. So it ends up, at the end of the day, a lot of the management is structured such that project managers and CIO's are rewarded or punished based on their ability to deliver projects on time.[2.1]

A relative new-comer to *the Bank* was also speaking about how he saw project leaders and CIO's being rewarded within *the Bank*:

a lot of the management is structured such that project managers and CIO's are rewarded or punished based on their ability to deliver. So they apply that pressure downward. Management and business are probably the most in-flexible areas. Because they're very much used to business having a lot of control. They have a lot of power, because they hold the money. So business sometimes do hold development to ransom, so to speak. [6.1]

The same analyst described the power [A-POWER] relationship between business and IT as uneven.

This organisation has got a really lean cost model. We [developers] all hate it because it's a means where we can get shafted. The business likes it though…[6.1]

And, in terms of making changes [A-CHANGE] to the methodology, the same analyst when describing how his project team was trying to update the methodology to incorporate object-oriented procedures, found further resistance [A-POWER] from business clients:

We attempted to modify the requirement specification to make them more like RUP. We wanted to include some of the RUP in the templates. But we had to check it with the business. And then business came back and said they didn't like some of the concepts, like use case diagrams. So we had to remove some of those key parts of the templates because business found them too confusing which is crazy, as it's a pretty simple concept.[6.1]

What is interesting in the above excerpt is that the business client is expected to learn the analysts' view of the world and speak the analysts' language, even though the excerpt makes it clear that the developers were simply trying to update the methodology. We found that in much of the interviews, when asked about documentation tools used to communicate design decisions with clients, they were expressed in terms of data models, process models, use case diagrams, etc. This documentation, which may not be technical in the sense of computer jargon, is technical in the sense of it being expressed in abstract concepts and terms with specific meanings that are foreign to the business client. The alien nature of the diagrams and the rejection of them by the business client added another layer of frustration to the systems developer. Another interesting observation was how the development side referred to the business client. The dichotomy between the systems developer and business client is evident in the interviewee's reference to the business client as "business" rather than as, for example, "clients", "partners", or "domain experts". The term "business" as used by developers connotes superiority: one who consumes, controls, prescribes and manages. From the perspective of developers, power [A-POWER] was seen to be vested with the business client. Asked specifically who drives systems development, a senior analyst responded in a way that was representative of many similar comments:

It's the business, definitely. Sometimes the IT areas will, once they get a project, try to drive what they think. But on the whole, the business are paying [for services and products] and whatever they want, gets done [7.3]

The same senior analyst even admitted that some project managers are scared of the power [A-POWER] that business wields:

I have worked for managers where they have agreed to deadlines that are too close and not reasonable. Sometimes too, I think they get a little bit scared of business. Like if the business wants something and they're demanding it, they're scared to tell them that their request isn't reasonable. [7.3]

However, one project leader described the opposite situation (a dissenting voice) when referring to the role of documentation and developers using it as a legitimate defence against business power [A-POWER]:

in terms of power, if you have signed it off you actually then basically have the power. It's a power thing where we don't do anything now unless we have their signed business requirements. So it is used, especially the business requirements, developers use them as a tool of power. Saying " you signed off this business requirement so this is what you want, so this is what we're delivering. You didn't ask for something else, you asked for this". So that's a power scenario. [9.1]

Based on a tactic to ensure credibility of interpretations known as member checking, and a focus on the most prominent codes, we feel confident that the following conclusion regarding sign-off and power is both authentic and contextually valid. The above excerpts are saying that the development life-cycle and sign-off process embedded within the ISDM creates a mechanism for the business client to exert and maintain power over the systems development group. Power and control is enacted as follows. Seeking sign-off and approval formalises the dependence of the system developer on the business client. Business will only 'sign-off' on the specifications and deliverables and take responsibility for project outcomes if they are satisfied that the system will meet their needs. In turn, business commit themselves to fund the next phase of development. Without sign-off, in principal, development cannot continue. Business also has the option to procure software services from an external third party. The developers are therefore dependent on gaining 'business' internally and accordingly often under-estimate costs, and agree to unrealistic schedules. Developers, on the other hand, seek to cover their position for any upstream errors that may require major re-work at a later stage, by seeking the business client to sign off on specifications and in effect gain agreement that the specifications are accurate and to accept responsibility for project outcomes.

Environment

By focussing outside the organisational, the *environment* framework draws attention to stabilized and institutionalised practices that take place within *The Bank*. According to the *social actor model*, the environment exerts technical and institutional practices (standards) on IT developers and *The Bank*. Lamb (2006) assert that the nature of the environment an organisation operates in greatly affects enactment of ICTs, and cites examples like

quality management practices that can be coerced through regulatory agencies, professional codes of ethics, laws, and industry-wide practices.

The code [E-STAND] was most prominent in the excerpts, indicating that all interviewees understood that the ISDM was mandatory, provided a standard for the collection and representation of data and in the ways it can be communicated. Many developers viewed the ISDM as a co-ordinating, controlling and project management mechanism by the controlling body – the business client. Employing the ISDM was supposedly mandatory, and its enactment was part of work culture. All systems work (except urgent fixes) had to adhere to it and the ISDM co-ordinates activities, regulates outcomes, and imposes standards [E-STAND]. However, as much of the focus within the IT department was on maintaining the many large legacy systems that support business processes and internal clients, most of developers work involved applications that are maintained and operated on large mainframe computers. The majority of these applications were developed using the traditional systems development life cycle in a lock-step fashion that conformed to the original intent of the in-house methodology. Many developers are still involved in the on-going maintenance and support of these legacy systems; however they view the methodology as embedded in their day-to-day duties, and in fact see the methodology as structuring their work. Most enact it uncritically, probably because they're told to follow it, not because they appreciate exactly why they're doing it. They're doing it because it's there, it's part of the culture of the organisation, so they enact it. According to the *social actor model* we need to recognise that enactment is institutionalised within *The Bank*.

Identities

The *identities* dimension describes the declared presentation and visible identity of individual systems developers as organisational members. From the transcripts, a range of identity factors were identified and coded according to the coding scheme. The identity factor identified as most influential was [ID-USE] the enactment of the methodology defines (some of) their identity as an IT worker. Systems developers enact ISDMs to create a positive image to construct identities and to control perceptions. Many programmers commented that the methodology helped define their legitimacy [ID-LEGIT] as a professional IT worker. In terms of the systems developers, knowing how to use the methodology and enacting the methodology competently can construct identities, legitimise their role, and control perceptions that they're professional. There were multiple data points confirming that the enactment of the methodology legitimises their role as an IT worker in the eyes of a project manager or the business user client.

To conclude the findings, the interviews are saying that in the end it is the business client who bears the most responsibility for the system – reviewing it, validating it, and signing off on it. The developers suggest that the client's role is secondary to theirs: almost a distraction to their work. Even so, the above transcripts are clear about who is really in charge of the development process. The business client has the domain knowledge, even though the systems developers do the thinking and apply their technical expertise. However, as suggested by Beath and Orlikowski (1994) systems developers are dependent on the business client in two main ways. Firstly, as noted earlier, systems developers need the business client to agree to build on existing technical and data infrastructures, and importantly to fund the design and construction of new or enhanced systems, and to cooperate with developers if budget and schedule commitments are to be met. Second, developers are dependent on the business client to validate and legitimate their contributions to the organisation, because if developer's interests are not acknowledged they have no organisational legitimacy [ID-LEGIT]. Hence, systems developers pursue their interests indirectly and subserviently by following the systems development process, leading and coaching the clients, and invoking 'directives' prescribed by the methodology.

Discussion of Findings, and Conclusion

In this paper, our particular concern is with the views of systems developers enacting a local and mandatory ISDM in their workplace, and the role of the ISDM as an agent of social and organisational reproduction. In conventional models of systems development, there is relatively little concern about understanding the relations of power that impede as well as enable the process of systems development.

The case depicts business exercising nearly complete control over the development process and systems developers as playing a reluctant, but submissive role. This finding is in direct contrast to the deconstruction of text of the Information Engineering (Martin, 1990) systems development methodology (Beath and Orlikowski, 1994) and other findings about the distribution of power, control and responsibility between systems developers and the business

client. These findings also shed new light on understanding the recurring conflict between developers and business users during systems development, which have been documented in the recent literature (Iivari and Huisman, 2007). The findings are significant in that they draw attention to the role of systems development methodologies and the control structures that are embedded in the methodology. As reported in the excerpts, the structures while not undetected by developers remain largely un-discussed and unexamined. Many interviewees when asked if they discussed the relative merits of the methodology with other colleagues said they did not.

The case also illustrates the active role of social institutions in the development of IS and the utility of the concept of user as a social actor. The case shows that pre-existing structures play an active role in constraining and enabling human agency in the use of an ISDM. The excerpts portray the business client as protecting their sphere of activity while developers are not seen as protective of their interests, and are relatively silent on the power issues which concern them. In the interviews, project managers did not lobby for the latest database technology, a more up-to-date methodology, or newer set of tools or techniques, or reusable code. Systems developers, at least in the job levels represented in the interviews, appear to have no parochial interests. Elkjaer et al (1991) in their examination of an in-house software development methodology found a similar ideology in the presentation of systems developers as free from political agenda.

Why then, is the business client dominant and the systems developer dependent – a mode of systems development not typically defined in the literature on systems development? The answer lies partly in the structural elements of the institutional context such as the division of labour, forms of control, locus of technological and work expertise, and importantly the allocation of resources and responsibilities mandated by the ISDM. The second half of the answer to why systems developers comply with an unequal power arrangement is that there is awareness on the part of developers that they are operating in a competitive labour market where there is pressure to perform and retain their well paid positions. With deregulation of the finance industry, intense competition among the finance sector locally and globally, and a recent down-sizing in the IT industry, some systems developers within *The Bank* were made redundant in a significant internal job-shedding exercise. By reflecting upon their work situation, systems developers are conditioned by a self-interested concern to secure and advance their position in a competitive labour market.

In reference to the utility of the concept of the *user as social actor*, we claim firstly that the four dimensions accord with SI literature (Kling *et al*, 2005:26), where the Bank's ISDM was conceptualised as part of a larger socio-technical ensemble. The sixteen coded behaviours drew focus on the way the ISDM was enacted. This ensemble view, according to Sawyer and Crowston (2004:52), is one where specific artefacts and people are interdependently connected through roles, uses of information, and actions. This conceptualisation of the ISDM and the various excerpts provide a good illustration of the so-called web models of IS in which people, their roles, their uses of a need for information, and their IT elements (ISDM) are connected together and embedded in a larger social milieu. The case also provides further evidence for the premise that the systems developer is best conceptualised as a social actor and not a technically focussed methodology user; and that there is an explicit link between individual actions and a larger social context.

All research designs have limitations. In our paper we have only examined one systems development methodology and thus we cannot generalise beyond it to others, although as Beath and Orlikowski (1994) comment, other ISDMs would mirror the same institutional contexts of systems development as described in this case because control over technical resources is tied to structures of power, meaning and norms. However, the findings are not suggesting that we can use the case as a 'test' of the framework. Rather, the case succeeds in establishing the plausibility of the *social actor model* by providing evidence of its capacity to provide meaningful analytical codes accounting for the enactment of the ISDM; and as argued, provides a fresh perspective on contextual issues surrounding ISDM enactment than reported on before. We have seen that our research provides an example of how institutional structures of *The Bank* provides an overarching, framing context within which systems developers often made constrained choices about methodology use. The case illustrates the constraining and enabling aspects of the methodology itself and the influence of the organisation on ISDM use. We concur with Lamb and Kling (2003) that their contextual model offers relevance in helping IS researchers to better understand ICT related phenomena.

Finally, we believe the findings discussed in this paper raise a central challenge for the field of IS. In general, both the research and practice of systems development have taken for granted rather than question the structural distribution of power, authority, control, and resources that constitute the institutional context of systems development. Understanding the context and roles that methodologies play in systems development will enable us to better understand the business client / systems developer relationship in the practice of systems development

References

Avison, D. and Fitzgerald, G. *Information Systems Development: Methodologies, Techniques and Tools*, (4th ed.) McGraw Hill, London, 2006.

Aydin, M. Harmsen, F. van Slooten, K. and Stegwee, R. "On the adaptation of an agile information systems development method", *Journal of Database Management*, 16(4), 2005, pp. 24-40.

Backlund, P. "Adopting the Knowledge Embedded in New Methods – The Challenges of Aligning Old and New Practices", *European Conference on Information Systems,* 2004.

Beath, C. and Orlikowski, W. "The Contradictory Structure of Systems Development Methodologies: deconstructing the IS-user relationship in Information Engineering", *Information Systems Research* 5(4), 1994, pp. 350-377.

Beynon-Davies, P. and Williams, M. "The Diffusion of Information Systems Development Methods", *Journal of Strategic Information Systems*, 12, 2003, pp. 29-46.

Chae, B. and Poole, M. "The Surface of Emergence in Systems Development: agency, institutions, and large-scale information systems", *European Journal of Information Systems*, 14, 2005, pp. 19-36.

Dhillon, G. "Dimensions of Power and IS Implementation"*, Information & Management*, 41, 2004, pp. 635-644.

Doherty, N. and King, M. "From technical to socio-technical change: tackling the human and organizational aspects of systems development projects", *European Journal of Information Systems*, 14, 2005, pp.1-5.

Elkjaer, B. Flensburg, P. Mourtinsen, J. and Willmott, H. "The Commodification of Expertise: the case of systems development consulting", *Accounting, Management & Information Technology*, 1(2), 1991, pp. 139-156.

Fitzgerald, B. Russo, N. and Stolterman, E. *Information Systems Development: Methods in Action*, McGraw Hill, Berkshire, 2002.

Gosain, S. "Enterprise Information Systems as Objects and Carriers of Institutional Forces: the new iron cage?" *Journal of the Association of Information Systems*, 5(4), 2004, pp. 151-182.

Goulielmos, M. "Systems Development Approach: transcending methodology", *Information Systems Journal*, 14, 2004, pp. 363-386.

Hall, R. *Organisations: Structures, Processes and Outcomes* (7th ed) Prentice Hall, 1999.

Hirschheim, R. and Klein, H. "Four paradigms of information systems development", *Communications of the ACM*, 32(10), 1989, pp. 1199-1216.

Iivari, J. Hirscheim, R. and Klein, H. "A Paradigmatic Analysis Contrasting Information Systems Development Approaches and Methodologies", *Information Systems Research*, 9 (2), 2001, pp. 1-30.

Iivari, J. and Huisman, M. "The Relationship between Organisational Culture and the Deployment of Systems Development Methodologies", *MIS Quarterly*, 31(1), 2007, pp. 35-58.

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

Jasperson, J. Carte, T. Saunders, C. Butler, B. Croes, H. and Zheng, W. "Power and Information Technology Research: a Metatriangulation Review", *MIS Quarterly*, 26(4), 2002, pp. 397-459.

Kautz, K. Madsen, S. and Nørbjerg, J "Persistent Problems and Practices in Information Systems Development", *Information Systems Journal*, 17, 2007, pp. 217-239.

Kling, R. Rosenbaum, H. and Sawyer, S. *Understanding and Communication Social Informatics: a framework for study and teaching the human contexts of ICTs*, Information Today, Medford NJ, 2005.

Langley, A. "Strategies for Theorising from Process Data"*, Academy of Management Review*, 24(4), 1999, pp. 691-710.

Lamb, R. "Alternative Paths Toward a Social Actor Concept", *Proceedings of the Twelfth Americas Conference on Information Systems,* Acapulco, Mexico, 2006, pp. 4113-4123.

Lamb, R. and Kling, R. "Reconceptualising Users as Social Actors in Information Systems Research", *MIS Quarterly*, 27(2), 2003, pp. 197-235.

Madsen, S. Kautz, K. and Vidgen, R. "A Framework for Understanding How a Unique and Local IS Development Method Emerges in Practice", *European Journal of Information Systems*, 15, 2006, pp. 225-238.

Markus, M, and Bjorn-Andersen, N. "Power over users: its exercise by systems professionals", *Communications of the ACM*, 30(6), 1987, pp. 498-504.

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

Nandhakumar, J. and Avison, D. "The Fiction of Methodical Development: a field study of information systems development", *Information Technology & People*, 12(2), 1994, pp. 176-191.

Orlikowski, W. and Iacono, S. "Research Commentary: desperately seeking the 'IT' in IT research – a call to theorising the IT artefact", *Information Systems Research*, 12(2), 2001, pp. 121-134.

Orlikowski, W. and Robey, D. "Information Technology and Structuring of Organisations", *Information Systems Research*, 2(2), 1991, pp.143-169.

Pickering, A. *The Mangle of Practice*, University of Chicago Press, Chicago, 1995.

Robey, D. and Markus, M. "Rituals in Information Systems Design", *MIS Quarterly*, March, 1984, pp. 5-15.

Rowlands, B. "The User as Social Actor: a Focus on Systems Development Enactment", *21st ACM Symposium on Applied Computing*, Dijon, France, 2006, pp.1540-1545.

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

Saunders, C. and Scammel, R. "Organisational Power and the Information Services department: a re-examination", *Communications of the ACM*, 29(2), 1986, pp. 142-147.

Sawyer, S. and Chen, T. *Conceptualising Information Technology and Studying Information Systems: Trends and Issues*, in Myers, M. Whitley, E. Wynn, E. and DeGross, J. (Eds) *Global and International Discourse about Information Technology*, Boston, Kluwer, 2002, pp. 109-131.

Sawyer, S. and Crowston, K. *Information Systems in Organisations and Society: Speculating on the Next 25 Years of Research*, in Kaplan, B. Truex, DP. Wastell, D. Wood-Harper, AT. DeGross, J. (Eds.) *Information Systems Research: Relevant Theory and Informed Practice,* Boston, Kluwer, 2004, pp. 35-52.

Schwandt, T. *Dictionary of Qualitative Inquiry*, 2nd Ed, Sage, 2001.

Sillince, J. and Mouakket, S. "Varieties of Political Process During Systems Development", *Information Systems Research*, 8(4), 1997, pp. 368-397.

Silva, L. "Epistemological and Theoretical Challenges for Studying Power and Politics in Information Systems", *Information Systems Journal*, 17, 2007, pp.165-183.

Scott, W. *Institutions and Organisations*, Thousand Oaks, CA, Sage, 2005.

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

Walsham, G. "Interpretive Case Studies in IS Research: Nature and Method", *European Journal of Information Systems*, 4(2), 1995, pp.74-81.

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