

2008

An Institutional Account of Systems Developers as Social Actors

Bruce Rowlands

Griffith University, b.rowlands@griffith.edu.au

Follow this and additional works at: <http://aisel.aisnet.org/sjis>

Recommended Citation

Rowlands, Bruce (2008) "An Institutional Account of Systems Developers as Social Actors," *Scandinavian Journal of Information Systems*: Vol. 20 : Iss. 2 , Article 2.

Available at: <http://aisel.aisnet.org/sjis/vol20/iss2/2>

This material is brought to you by the Journals at AIS Electronic Library (AISeL). It has been accepted for inclusion in Scandinavian Journal of Information Systems by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

The Enactment of Methodology

An Institutional Account of Systems Developers as Social Actors

Bruce Rowlands

School of ICT, Griffith University, Brisbane, Australia

b.rowlands@griffith.edu.au

Abstract. This paper reports on research into how systems developers enact a systems development methodology (SDM) with a focus on describing how method enactment is bound up in everyday social and institutional structures. The case study develops the argument that institutional structures (such as authority, norms and routine ways of doing things) embedded within the methodology are active forces in the systems development process. We ground our argument on the findings from a study of an in-house developed SDM in a large IT department within a major financial institution in Australia. The findings show that despite the rhetoric of business client involvement working in unison with systems developers, the excerpts depict a conflict of interests with the client exercising nearly complete control over the development process and the in-house developers playing a submissive role. In terms of contribution to research, the study operationalises a theoretical framework integrating elements of a social actor model outside its original domain to provide a deeper understanding of the institutional forces at play in information systems development.

Keywords: SDM enactment, social actor model, institutional structures.

1 Introduction

This paper reports on field research into how systems developers enact a systems development methodology (SDM) in their everyday work practices. At the technical and organisational level the enactment of SDMs is a topic of significant research attention within information systems (IS) (Iivari et al. 2001) especially in view of the well documented costs associated with the development and maintenance of an aging application portfolio. One solution to costs is seen in terms of adherence to methodologies to plan and rationally control the development and maintenance process. Textbooks and considerable IS development research implicitly assume that methodologies are used and are useful (Avison & Fitzgerald 2006). On the other hand, the usefulness of SDMs has been questioned because of their lack of practicality (Fitzgerald 1997, 1998; Wynekoop & Russo 1997), that they offer transient regularities in work practices, and that they artificially structure and interfere with the development process (Truex et al. 2000).

Despite the divergence of opinion between the intent of methodologies and concrete practice, researchers still have an incomplete understanding of how practitioners collectively use methodologies in their day-to-day work or the forces that impact on the situated use of methodologies (Iivari & Huisman 2007). The work that has been carried out and published is limited in its ability to consider the social and institutional aspects of methodology use. It is our contention that most field research on methodology enactment, (with the noted exception of Backlund 2004, Bansler & Havn 2004; Kautz et al. 2004; Aydin et al. 2005; Madsen et al. 2006; Iivari & Huisman 2007) has an individualistic, systems developer orientation with little recognition that software development is improvised, emergent, with unanticipated outcomes taking place in a complex social context (Bansler & Havn 2004). Similarly, there has been little consideration of the authority relations that exist in the client-developer relationship, especially when the client and developer come from the same organisation.

A lack of field research on the impact of authority relations and institutional forces precludes a full understanding of how SDMs are enacted. Researchers have long called for research on methodologies in real life organisational situations (Wynekoop & Russo 1995) and they continue to do so (Kautz et al. 2007). With this call in mind, the purpose of this paper is to investigate the relationships among systems developers, the business client, the SDM, and context that infuses meaning and purpose surrounding its use. We need to shed light on how systems developers' concerns and every day work practices are shaped by institutional structures including the SDM itself and the authority relations existing between the business client and systems developers. This paper argues that SDM enactment needs to be understood

in a wider institutional context comprising both social relations and social structures in and outside the organisation. Accordingly, our question is, how is enactment influenced by organisational and institutional context?

The paper answers this question by operationalising a framework that integrates elements of a social actor model (Lamb & Kling, 2003) to provide a deeper understanding of the institutional forces at play in information systems development. Insight with respect to this focus is derived from a case study of the deployment of an in-house developed methodology in a large IT department of a major Australian bank (*The Bank* – a pseudonym).

Following Iivari & Huisman (2007), this paper uses the term methodology very broadly to cover the totality of systems development approaches, while Fitzgerald et al. (2002) also refer to methodologies in a broad way as any formally documented in-house or commercially available systems development approach. Employing a broad interpretation is necessary due to the difficulty in defining and measuring SDM 'use' (Iivari & Huisman, 2007). To overcome this ambiguity, Iivari & Huisman (2007:38) prefer the term 'deployment' rather than 'use', and distinguish three major aspects of SDM deployment: use, impact, and support. Methodology *use* refers to the intensity of methodology deployment across an organisation, while methodology *impact* refers to aspects of quality and productivity issues in systems development. In this paper, we adopt the latter meaning in terms of a methodology's *support* role: as a production, coordination, and organisational technology.

Two further key terms as used in this paper need defining. An institution is defined as any standing, social entity that exerts influence and regulation over other social entities as a persistent feature of social life (King et al. 1994, p. 141). Structure refers to the form of social context and material conditions which define the range of actions that are available to social actors (Grix 2004, p. 49). Using this definition, institutions are structures that impact on individual and group work practices.

This research is distinct and important for the following reason. Previous research on SDMs tends to focus on the features of the methodology and systems developer's behaviours while underemphasising the role of context and institutional structures (Chae & Poole 2005). Among the many studies of SDM enactment, few pay attention to the role of context and the institutional structures embedded in a systems development methodology. The intent in this case is to contribute towards developing socio-organisational theory about SDMs in institutional and cultural contexts.

2 Theoretical framework

The paper proposes a social informatics (SI) perspective as a useful lens through which to investigate how systems developers enact their local methodology within projects. A social informatics view argues that SDMs and people can best be examined 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. A further assumption of SI underpinning this paper is that SDMs are more than just technical artefacts but are social institutions that exert their own type of agency, and that SDMs also interact with human agency in the systems development process.

Our attempt to understand the enactment of an in-house developed methodology in a large-scale organisation made us aware of a need for an analytical framework that addresses issues of the technological artefact, the role that actors play in enacting the technology, and at different levels of analysis. One of the most prominent lines of work is research on conceptualising the user as a social actor (Lamb & Kling 2003; Lamb 2006). Drawing on the work of Scott (2001), Lamb (2006) describes how the social actor concept has been theoretically supported by institutional approaches, whereby institutions provide a framing context within which social actors make constrained choices about ICT use, particularly when they are situated in organisations.

Lamb & Kling's (2003) framework focuses on the use of ICTs, and aims to develop a social actor model of 'users' in an organisational setting. The ICT in this framework, is seen as one (among several) organisationally embedded artefacts that interact with 'users' and shapes their work. SDMs are also organisationally embedded artefacts that are applied in complex organisational settings, where other influences – such as institutional structures and context – also play a role. This paper argues that we can transfer Lamb and Kling's framework from one setting (use) to another (development) substituting SDM for ICT. For instance, there are several parallels between ICTs and SDMs: they are technologies, are both artefacts that can (and do) regulate the work of their respective 'users', and there are many similarities in enactment. In modern organisations, for example, 'users' of technology often circumvent their job tasks through a process known as technology appropriation, defined as the way that users evaluate and adopt, adapt and integrate a technology into their everyday practices (Carroll et al. 2002). Similarly, developers can (as illustrated in the case) build systems without paying too much attention to all aspects of the SDM. Furthermore, the work situation has parallels. Systems developers use the technology to build a product or a service—documentation or systems. Users in the original domain (Lamb & Kling 2003, p. 230) came from the pharmaceutical,

law, and real estate industries where the work practice involved professionals using ICTs to construct services and products online. This transfer of ICT to SDM enables us to demonstrate an application of the social actor model outside its original domain. To avoid potential confusion, we need to point out that in this example involving systems developers, it is the developer who is the ‘user’ of the technology (the SDM), in contrast to conventional IS literature, where the user is often portrayed (let’s say) as an office worker being the recipient of a developed system.

According to Lamb & Kling’s (2003) *social actor model*, people’s individual autonomy and their behaviours are shaped by the social norms (dress standards, working hours etc), institutional forces (routine work practices), and the social and physical structures that surround them. This approach reflects what Orlikowski & Iacono (2001) identify as the ‘ensemble view’ of technology where ICTs are components of a more complex socio-technical ensemble that include people, work practices, and institutional and cultural factors. Specifically, this research explores how enactment of a particular SDM can be understood by reconceptualising the systems developer as a social actor. Using Lamb & Kling’s (2003) argument, systems developers (social actors) are not simply users (or operators) of a SDM, but rather their actions (their agency) are channelled through a complex, multilevel system of networks and organisational affiliations.

In this paper, we therefore define *enactment* of the SDM 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 SDM itself. Explicit objects produced by systems developers during 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 of *enactment* as interplay between context and developer intentions, the case analysis applied the Lamb & Kling (2003) *user as social actor* model as a conceptual lens to understand methodology enactment within a single organisational setting.

The *user as social actor* model involves four dimensions as shown in Table 1—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 within projects. The second two dimensions—*affiliations* and *environments* relate people to their organisation, and to the industries and environments of those organisations. 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 (Lamb, 2006). According to the *social actor model*, the user is not a technically focused or socially thin, passive user of technology. The user is a person who acts with information technology in a social setting.

<i>Social Actor Dimensions</i>	<i>Characteristics and Behaviours of Connected and Situated Individuals (Lamb and Kling, 2003, p. 213)</i>
Identities (Definition: avowed presentations of the self and ascribed profiles of organisation members as individual and collective entities).	Social actor identities have an ICT use component.
	ICT-enhanced networks heighten multiple identities as expert or novice.
	ICT-enhanced connections among organisation members transcend roles.
	Social actors use ICTs to construct identities and control perceptions.
Interactions (Definition: information, resources and media of exchange that organisational members mobilise as they engage with members of affiliated organisations).	Organisational members seek to communicate in legitimate ways.
	Organisational members build, design and develop interactions that make information actionable.
	ICTs become part of the interaction process as people transform and embed available informational resources into connections and interactions.
	As organisational members, people perform socially embedded (role-based), highly specialised actions on behalf of the organisation.
Affiliations (Definition: organisational and professional relationships that connect an organisational member to industry, national and international networks).	Social actor relationships are shaped by networks of organisational affiliations.
	Relationships are dynamic, and related informational exchanges, they change with flows of capital, labour, and other resources.
	Relationships are multilevel, multivalent, multi-network ie. local/global group, they exert influence on organisational, inter-group, inter-organisational cultures.
	As relationships change, interaction practices migrate within and across organisations.
Environments (Definition: stabilised, regulated and/or institutionalised practices, associations and locations that circumscribe organisational action).	Organisational environments exert technical and institutional practices (standards) on the company and their members.
	Environmental dynamics require a display of overall competence.
	ICTs are part of the organisational environment.
	ICTs are part of the industry, national, and/or global environment.

Table 1. Multi-dimensional conceptualisation of a social actor

To illuminate the institutional context that impacts on methodology enactment, we provide Lamb & Kling's (2003) general description of the four interdependent social actor dimensions as shown in Table 1. Firstly, social actors continually improvise their uses of the ICT and reconfigure their organisational roles (identi-

ties) to reconstruct and represent themselves as competent, ICT-savvy social actors. Secondly, their interactions are (mostly) legitimate, action-enabling, constructed and role-based. Thirdly, affiliations are networked, exchange-related, multiple and changing. And finally, their environments are technical, institutional, ICT-enhanced, and expansive. In sum, social actors are not just ‘users’ of ICTs—they effectively re-design ICTs in use (Lamb 2006, p. 4115).

Each of the four dimensions of the *social actor model* in Table 1, are further comprised of four behaviours. These sixteen characteristics and behaviours of connected and situated individuals were used to develop the initial coding scheme (cf. Tables 2, 3, 4, & 5) for the qualitative analysis of data (to be discussed in Section 4).

3 Research approach

The research centred on exploring the impact of institutional structures on the enactment process and how systems developers carry out their work in The Bank. Thus, Yin’s (1994) definition of scope for a case study is applicable as the boundaries between the phenomenon—the SDM defined in a *support* role—and the context of enactment are not clearly evident.

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 SDM is important because systems developers with different interpretations will enact the SDM differently. Secondly, while the use of the methodology in *The Bank* was mandatory, the researcher did not seek to privilege the position of methodologies in information systems development (ISD). Keeping an open approach in line with Truex et al. (2000) who question the assumption that systems development is a methodical process was important so as to not inhibit our understanding of how information systems are developed or more specifically how the SDM was enacted in practice. Therefore, an interpretive case study (Walsham 1993) was chosen to produce a subjective understanding of phenomena from both the researcher’s and systems developer’s perspective.

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 ICT 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 SDM. Importantly, the banking industry is highly technical, highly competitive, highly regulated and institutionalised.

The source of data for the case study comprised interviews with individuals from one occupational community: IT professionals at various levels (programmers to senior project managers) all within the systems support, development, and method

support divisions. The unit of analysis operated at various levels, however because systems developers are the enactors of the SDM and developers of the information systems, it is the influence of institutional structures on their intentions and actions that are the centre of attention. In terms of this case study, *institutional structures* refers to work procedures mandated by the SDM such as gaining sign-off before commencing the next stage of development, the day-to-day interactions within and among project groups, and authority based on power and expertise among the bank's IT department. Given this view of structure, systems developers can be seen as complex social actors acting in constrained ways, rather than simple 'users' of the SDM (Lamb & Kling 2003).

To enhance the credibility of the interview process, all semi-structured interviews were guided by an interview protocol and dealt with the following general 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. It needs to be pointed out that we did not initially ask specific questions, instead we deliberately kept the case interview questions open leaving the developers to tell us their story about what influenced them in their use of the SDM. Later, we used the social *actor model* as a theoretical lens to draw meaning from the interviews. 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, p. 155). The analysis of transcripts involved five steps. The techniques adopted during each step are explained below.

Step 1: *Interpretive translation of the original model*. In this step, the sixteen characteristics and behaviours of connected and situated individuals (as shown in Table 1) were interpreted by the author and translated to an equivalent meaning in terms of the case scenario. For instance, in the social actor dimension *Identity*, the first behaviour is that “the social actor identities have an ICT use component” (cf. Table 2, column 2). The author had already conducted the interviews and had re-read the transcripts a number of times. By being familiar with the context, the author interpreted the first behaviour to mean “the enactment of the SDM defines (some of) their identity as a systems developer” (cf. Table 2, column 3). While multiple interpretations were possible, and through a process of iteration and revision over the transcripts, the author settled on an interpretation that best fitted the meaning of the original construct. This process of interpretation and translation of the original constructs to a match in the transcripts continued for the entire sixteen behaviours of the connected and situated individual.

Step 2: *Assigning codes*. Based on key words from the original user as social actor model (as described in Table 1), codes were assigned to match the examples from the case (cf. Table 3, column 1). An example of code assignment is taken from the author's interpretation from the transcripts, as described in Step 1. For instance, the four behaviours in the social actor dimension *Interaction* in Table 3 were given the prefix IN. The last behaviour in this dimension is that "as organisational members, people perform socially embedded (role-based), highly specialised actions on behalf of the organisation". Following Step 1, this was coded as IN-CONSTR based on the author's interpretation from the transcripts that "using the SDM dictates and CONSTRains their role within the bank – it tells developers what they must do – it is the bank's way". Following this format, the author created and operationalised a list of codes prior to content analysis (step 3) based on the user as social actor model presented in Table 1. These codes are listed in columns 1 of Tables 2, 3, 4, & 5 respectively.

Step 3: *Content analysis*. Once text were collected and transcribed each of the interviews were imported into a data management software tool (File Maker Pro v5). The *social actor model* was used in a form of content analysis (Schwandt, 2001, p. 34) where the text was systematically listed and coded according to the sixteen behaviours of the connected and situated individual. The list of researcher-constructed codes that best captured the description of the phenomenon was then deductively applied to the text to codify, count, and extract the data. This same format was carried through the entire thirty interviews.

Step 4: *Managing the data*. In terms of data management issues, the process of analysis was assisted by and recorded in a database through procedures such as importing transcribed interviews, coding against the sixteen constructs, adding comments and reflections, sorting the interpretations by code; and text retrieval of selected instances into the body of the research report.

Step 5: *Writing the case*. 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 project manager involved in the use of the SDM. 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.

4 Case Analysis & Findings

4.1 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. 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, etc.), investment banking (shares, fund management) and internal accounting (general ledger, payroll, etc.)

4.2 The SDM and Practice

To develop, customise or maintain these systems, the IT division has developed and documented an internal systems development methodology applicable for all development and maintenance tasks. The methodology, known as the SDM—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. 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. For many systems developers, knowledge of the SDM was acquired on the job and internalised over time. In reality, method use is implicit and for most an unconscious process in which knowledge is inter-twined with practical experience.

The methodology itself does not mandate particular tools or techniques. The scope of the SDM 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. Projects are initiated by the business client in all instances. The business client may wish to create a new system, or change existing systems to comply with regulations. This initiation happens within a team in a business unit. The business client engages with their management to get approval and funding, and if approved, a business project manager is appointed and the project commences with the customer (the business client) securing IT services from the supplier (in most cases) the in-house IT department. The systems development process is commenced once the project is initiated and an IT team appointed. The deployment of the SDM use usually starts with the business requirements gathering and analysis phase. While the use of the SDM is mandatory within the Bank, deployment varies, and is dependent on contingencies such as: the business client's expectations and knowledge of the project, their prior experience with the SDM, the IT project manager's preferences and skills, the size of the project, time and budget constraints, and as commonly documented in the literature—developers' knowledge and skills in methodologies and their willingness to try new processes.

The size of the project will determine which documents are required and their content. The IT project manager can check the *Quick Reference Guide* on what documents are required and will schedule them into the project plan. Similarly, when planning the actual phase in detail he/she appoints the resources for completing the document. Once a document is produced it is sent for a review to the distribution list identified by the project manager. Some are needed to sign-off or review the document; others just receive the document for information purposes. Quite often, there are formal document review sessions where key project members from the business-client, the hardware vendor, and developers gather with the author of the document (from the development team) to 'walk through the document' before the sign-off deadline date. This process is more likely to happen in the business requirements and detailed design phases. The author of the document follows-up by calling for signatures for the sign-off of the document. Usually these sign-off are in a form of email. Once the feedback is incorporated into the document, the document is placed into the project repository on *The Bank's* intranet where all other current and former project related documents are stored.

Given that all projects are funded in phases, some of the mandated documents are required as completed deliverables as proof to gain funding for the next stage. Important phase documents include a *Project Concept and Analysis*, a *Software Requirements Specification* and a *Detailed Design Review*. Members of the support team are also required to sign-off on the detailed design, as they will be responsible for the system while in production. Most of the emphasis is on the earlier phases of the lifecycle with less emphasis on the build, testing and installation phases. In theory, development cannot start until the detail design review document is signed

off. In practice, usually some development work starts before the final sign-offs depending on the judgment of the IT project manager.

4.3 Social Actor Analysis

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 & Sarbannes-Oxley). What follows is an application of the model to the specific characteristics of the case.

4.3.1 Identities

<i>Social Actor Dimensions and Code</i>	<i>Characteristics & Behaviours of Connected and Situated Individuals (Lamb & Kling 2003, p. 213)</i>	<i>Examples from the Case</i>
[ID-USE]	Social actor identities have an ICT use component	The enactment of the SDM defines (some of) their identity as a systems developer.
[ID-COMP]	ICT-enhanced networks heighten multiple identities as expert or novice	Knowledge & competent enactment of the SDM can define the developer's identity as novice or expert.
[ID-HYBRID]	ICT-enhanced connections among organisation members transcend roles	The SDM can be used differently according to roles they occupy within the project group.
[ID-LEGIT]	Social actors use ICTs to construct identities, legitimise their role, and control perceptions	Enactment of the SDM legitimises their role as a systems developer IT worker in the eyes of the business user client or project manager.

Table 2. A Social Actor View of *Identity*

From the transcripts, a range of identity factors were identified and coded according to the coding scheme shown in Table 2. The identity factor identified as most influential was [ID-USE] the enactment of the methodology defines (some of) their identity as a systems developer. For instance, systems developers enact the methodology 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 systems developer and made them feel assured as others were happy with their work. For example, one of them said:

For me personally I'm comfortable with the way I do my work and people I work with haven't had a problem with it. So I guess, approach wise, I'm doing the right thing. And in terms of compliance with the methodology I haven't hit a problem with that in my years of working here.

In sum, the *identities* dimension describes the declared presentation and visible identity of individual systems developers as organisational members. Systems developers regularly enact SDMs to compile and present information to various affiliates. In so doing, they create an identity for their organisation and for themselves.

4.3.2 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 SDM 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. The analysis identified a range of interaction behaviours such as producing documentation, communicating, and acting in constrained ways.

According to the social actor model organisational members seek to communicate in legitimate ways. In its most visible form, the SDM 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

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 reasons for not wanting to comply with the methodology. Many developers felt that documentation [IN-DOCN] often overwhelmed the business client, but agreed that for developers, detail was necessary. Even the process of writing the documentation was regarded

as an impediment to documenting the system. This was a common complaint among developers because they are not writers but felt they are simply here to build systems.

<i>Social Actor Dimensions and Code</i>	<i>Characteristics & Behaviours of Connected and Situated Individuals (Lamb & Kling, 2003, p. 213)</i>	<i>Examples from the Case</i>
[IN-DOCN]	Organisational members seek to communicate in legitimate ways	The SDM mandates documentation throughout all phases & calls for meetings, both formal & informal among affiliates to review them.
[IN-ACTION]	Organisational members build, design and develop interactions that make information actionable	The SDM mandates the generation of specifications becoming actionable documents requiring a sign-off at each stage. This is the work culture imposed by business clients on developers.
[IN-TAILOR]	ICTs become part of the interaction process as people transform, tailor and embed available informational resources into connections and interactions	Method in use, the developer tailors the SDM in pragmatic and ingenious ways as part of the design documentation process.
[IN-CONSTR]	As organisational members, people perform socially embedded, highly specialised actions on behalf of the organisation	Using the SDM dictates and constrains their role within the bank – it tells developers what they must do – it is the bank’s way.

Table 3. A Social Actor View of Interactions

While there were many mentioned down-sides to producing the documentation than positives many commented that the technical details captured can help later on 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 also saw the SDM 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.

However, the main reason for seeking to communicate in legitimate ways was to get documents [IN-DOCN] signed and gain approval to commence the next stage of development. A further example where organisational members sought to communicate [IN-COMM] in legitimate ways was the use of common terms (based on deliverables) that the SDM 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 SDM provided a terminology so when systems developers talked among themselves people knew the language of the business. In other words, the methodology provided a common language and was consistent.

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 organisational members to develop interactions that present information in an acceptable standardised format. The *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 the sign-off in two ways: positively—to gain approval so that work could commence on the next stage; and negatively—it is 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 as a major functional component of the methodology, and a major reason for having a formal software development process. A project manager commented:

...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 SDM] forces you to do things.

The above excerpts illustrate that sign off is a work structure imposed by the business client. The SDM mandates the generation of specifications becoming actionable documents [IN-ACTION] requiring sign-off at each stage. The interviews also identified, especially among new development teams, a feeling that the existing SDM was obsolete and due for an overhaul. Asked what would be required to introduce an entirely new methodology with the existing done away and phased out, the method support manager replied: ... *it'd be a big effort*.

To introduce a new SDM would involve serious change management initiatives and this would also involve gaining business agreement. 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 people 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.

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 practices and routine patterns of work. Using the SDM 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 is the bank's way thereby enforcing the rules, regulations of the bank, and the politics 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:

.. developers don't have a lot of say. 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. That's what I don't like about it.

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%. 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.

The above situation provides another 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 the generation of poor quality code). The prominent interactions [IN-ACTION] and [IN-CONSTR] illustrate that sign-off is a work structure imposed by the business client. The SDM mandates the generation of specifications requiring sign-off at each stage. From an authority perspective, the life-cycle, sign-off, and routine

patterns of work embedded within the SDM create a mechanism for the business client to exert and maintain control over the systems development group. The excerpts also illustrate a dichotomy of mindset between the systems developer and the business client.

4.3.3 Affiliations

According to Lamb (2006) networks are a basic configuration for organising social, economic and political exchanges. The *affiliations* dimension in Table 4 draws 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 developers 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 on a day-to-day basis within *The Bank* with business clients. These interactions bring about change [A-CHANGE], and through this change it shapes how the methodology is used, and in particular how it evolves 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), 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 developers:

Business sometimes 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.

<i>Social Actor Dimensions and Code</i>	<i>Characteristics & Behaviours of Connected and Situated Individuals (Lamb & Kling, 2003, p. 213)</i>	<i>Examples from the Case</i>
---	--	-------------------------------

[A-NET]	Social actor relationships are shaped by networks of organisational affiliations	Multi project teams including hardware & telecommunications partners have a mandated role in verifying specifications.
[A-EXCH]	Relationships are dynamic, and related informational exchanges, they change with flows of capital, labour, & other resources	There was down-sizing in the IT industry generally, and a tight IT labour market. The bank often used contractors, and outsourced aspects of IT work.
[A-POWER]	Relationships are multilevel, multivalent, multi-network i.e. local/global group, they exert influence on organisational, inter-group, inter-organisational cultures.	Project managers & team members are required to deal with business clients from various sections (social networks) within the bank to complete the project, however power is vested with the business clients.
[A-CHANGE]	As relationships change, interaction practices migrate within & across organisations	Developers 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.

Table 4. A Social Actor View of *Affiliations*

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. Business are used to 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.

This analyst described the power [A-POWER] relationship between business and developers 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.

The above excerpts provide further evidence of a clash of interests where the business client is 'holding development to ransom', and 'applying pressure downwards'; whereas according to developers, both parties 'should be working together'. Another interesting observation from the excerpts was how the development side referred to the business client. The conflict of mind-set 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 is 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.

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 are demanding it, they are scared to tell them that their request isn't reasonable.

The above excerpts are saying that contracts, the development life-cycle and the sign-off process embedded within the SDM create a mechanism for the business client to exert and maintain power over the systems development group.

4.3.4 Environment

According to the *social actor model*, the environment an organisation operates in is formed by the kind of affiliations it has formed with industry, financial institutions and its clients. To understand methodology enactment we need to recognise the regulated and/or institutionalised practices of *The Bank*, and other associations that circumscribe organisational action. By focusing outside the organisation, the *environment* framework in Table 5 draws attention to stabilized and institutionalised practices that take place within *The Bank*.

<i>Social Actor Dimensions and Code</i>	<i>Characteristics & Behaviours of Connected and Situated Individuals (Lamb & Kling, 2003, p. 213)</i>	<i>Examples from the Case</i>
[E-STAND]	Organisational environments exert technical and institutional practices (standards) on the company and their members	SDM enactment is supposedly mandatory. It is part of the work culture. All systems work (except urgent fixes) must adhere to it and the SDM co-ordinates activities and regulates outcomes.

[E-COMP]	Environmental dynamics require a display of overall competence	Placing the SDM on the Intranet creates an online presence displaying the project status and associated documentation. This is IT policy.
[E-INVEST]	ICTs are part of the organisational environment and require a substantial investment	The Bank has a significant SDM investment in personnel, development tools, techniques, and in-house training. There are rules, regulations & norms that specify how systems are developed.
[E-INFRA]	ICTs are part of the industry, national, and/or global environment and many software vendors invoke infrastructure richness that promotes use	Method support bemoaned the fact that other proprietary SDMs were fully online, and acted as a knowledge repository of design decisions, thereby promoting a positive use of an SDM.

Table 5. A Social Actor View of *Environments*

The code [E-STAND] was by far the most commonly stated in Table 5, indicating that all interviewees understood that the SDM was mandatory, provided a standard for the collection and representation of data and in the ways it can be communicated. Many viewed the SDM as a co-ordinating and project management mechanism by the controlling body—the business client. Employing the SDM was supposedly mandatory, and its enactment was part of work culture. All systems work (except urgent fixes) had to adhere to it and the SDM co-ordinates activities, regulates outcomes, and imposes standards [E-STAND]. According to the social actor model the *environment* and adherence to industry-wide work practices exert technical and institutional (standards) on *The Bank* and played a significant guiding role as one project leader put it:

With a mixture of skill sets it is important that we all have the same standards and the outcomes are the same. A common benchmark is good because of this, and it can be used as a guideline—and the methodology allows everyone to follow a particular guideline.

Even a Java developer who was not keen on the waterfall nature of the methodology agreed that with mobility within *The Bank*, using a common standard [E-STAND] made sense. Most within the bank understood this rationale and were accepting its purpose and without a standard set of work practices embedded within a methodology [E-STAND] a support analyst admitted work would be messy.

This opinion of conforming to a set of standards was strongly evident among the interviews. Many systems developers are still involved in the on-going maintenance and support of legacy systems. The majority of these systems were developed using the traditional systems development life cycle in a lock-step fashion that conformed

to the original intent of the methodology. The developers 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 are told to follow it, not because they appreciate exactly why they are doing it. They are doing it because it is there. It is 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*.

4.4 Findings

This paper sheds light on the nature of collaboration between developers and the business client during systems development in IT projects that cross intra-organisational boundaries. While supposedly a joint process, developers see it as an unequal process, with a conflict of interests in which the business client achieves their objectives to the relative disadvantage of developers.

Specifically, the findings portray systems developers as dependent on the business client in two main ways. First, as noted in the *interactions* section, systems developers need the business client to fund the design and construction of new or enhanced systems, and to cooperate with other developers if budget and schedule commitments are to be met. Second, as reported in the *identities* section, developers are dependent on the business client to validate and legitimate their contributions to the organisation. Knowing how to use the methodology and enacting the methodology competently can construct *identities*, legitimise their role, and control perceptions that they are professional. There were multiple data points confirming that the enactment of the methodology legitimises their role as a systems developer in the eyes of a project manager or the business client. Hence, systems developers pursue their interests indirectly by following the systems development process, leading and coaching the clients, and invoking ‘directives’ prescribed by the methodology.

From the *affiliations* and the *interactions* sections, the interviews are saying that in the end it is the business client who bears the most responsibility for the system – funding it, reviewing it, and signing off on it. However, in terms of ISD, the developers are suggesting that the client’s role is secondary to theirs—almost a distraction to their work as they focus on technical issues and system building. Even so, the *affiliation* transcripts confirm the inherent power of the business client – yet there is a dichotomy of mind-sets. The business client is more interested in controlling costs and monitoring deadlines, whereas the developer is more interested in building quality systems and employing their technical expertise.

The *affiliation* excerpts also tell us that it is the policies and practices embedded in the SDM through sign-off and stage-gate funding that constitute the struc-

tural exercise of power. The findings indicate 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. All these findings are significant in that they draw attention to the role of systems development methodologies, control structures, conflicting interests, and power that appear largely outside the domain of the systems developer.

5 Discussion

This study was motivated by the need to better understand how systems developers enact a systems development methodology (SDM) with a focus on describing how method enactment is bound up in everyday social and institutional structures. There is a need to know more about how systems developers' concerns and everyday work practices are shaped by institutional structures including the SDM and the authority relations existing between the business client and systems developers.

5.1 Implications for Practice

What are the practical implications of these findings? 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 & Bjørn-Anderson 1987). The findings give evidence that knowledge of the systems development process is not a key precursor of control. Arising from this is the following question. How is the business client able to maintain control over the systems development process even when they do not possess the level of understanding of ISD practices comparable to that of IT professionals? According to this case, the answer lies in the structural elements of the SDM and 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. Our case study specifically indicates that it is the policies and work practices embedded in the SDM that constitute the structural exercise of power by the business client over the systems developer.

Bjerknes & Mathiassen (2000) presented a similar argument and discussed the balance between trust and control with regards to contracts and client-developer relationships. In their case of projects using the Rational Unified Process (RUP), the business client was mainly interested in the adherence to milestones and the monitoring of progress according to agreements. In their case, the business client had tre-

mendous influence over the course of projects through the utilisation of milestones embedded in the methodology. Our case study supports this finding.

Our study also indicates that control structures embedded in the methodology, while not undetected by developers, remain largely un-discussed and un-examined. For instance, many interviewees when asked if they discussed the relative merits of the methodology with other colleagues said ‘they did not’. The case shows that pre-existing structures play an active role in constraining and enabling human agency in the use of an SDM. The excerpts also 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 SDM found a similar ideology in the presentation of systems developers as free from political agenda. Why would this be the case?

Hirschheim & Klein (1989) offer one such plausible explanation. These authors posit that all systems developers approach the development task with a number of explicit and implicit assumptions about the nature of human organisations, the nature of the design task, and what is expected of them. According to Hirschheim & Klein (1989) these assumptions play a central role in guiding the ISD process. For instance, when discussing systems development as a process of instrumental reasoning, the business client is portrayed as responsible for providing the systems objectives, the systems developer is the expert who takes the objectives and turns them into a constructed product — the system. The business client dictates the ends; the developers use specific means to achieve the ends. This portrayal of the IS developer is in accordance with a number of appealing and currently orthodox beliefs in software engineering: the developer is the expert in technology, tools, and methods of systems design and implementation. These orthodox beliefs render the issues of power, conflicting interests, and systems goals appear outside the domain of the systems developer.

Why then are these beliefs perpetuated, and why is the business client dominant and the systems developer dependent? As many of the excerpts show, and as discussed above, the answer lies partly in the allocation of resources and responsibilities mandated by the SDM. The second half of the answer, as 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, a recent down-sizing in the IT industry, some developers within *The Bank*

were made redundant in a significant internal job-shedding exercise. By reflecting upon their work situation developers are conditioned by a self-interested concern to secure and advance their position in a competitive labour market.

Finally, in terms of practice, managers and developers need to acknowledge that SDMs largely subsume assumptions, meanings, and expectations in an organisational context in which they are used and consumed. SDMs are not produced or enacted in a vacuum, and the conditions of the methodology's creation and appropriation shape its form, content and interpretation. A local SDM is both a derivative of its institutional context, (which includes industry-wide structures of professional competence, historical patterns of resource allocation, assumptions about who has control), and how and where work gets done.

5.2 Implications for Research

In terms of theory development, this case succeeds in establishing the plausibility of the *user as social actor model* (Lamb & Kling 2003) by providing evidence of its capacity to provide meaningful analytical codes accounting for the enactment of the SDM; and as argued, provides a fresh perspective on contextual issues surrounding SDM enactment than reported on before. We have seen that our research provides an example of how institutional structures of the bank provide an overarching, framing context within which systems developers often made constrained choices about methodology use. We concur with Lamb & Kling (2003) that their contextual model offers relevance in helping IS researchers to better understand ICT related phenomena.

In reference to the utility of the concept of the *user as social actor*, we claim that the four dimensions accord with SI literature (Kling et al., 2005), where the bank's SDM was conceptualised as part of a larger socio-technical ensemble. This ensemble view, according to Sawyer & Crowston (2004), is one where specific artefacts and people are interdependently connected through roles, uses of information, and actions in a larger social milieu. In terms of conceptualising the IT artefact, the focus has been on the ways that the institutional context shapes the on-going development, deployment, and use of the SDM. The findings for research are two-fold: the first is that systems developers are not just technical users of a systems development methodology – they perform many roles and activities in the workplace that have to be taken into account when understanding the enactment process. They liaise with the business client, the hardware vendor, contractors, other software houses, project members, and with other internal IT sections. This finding supports Lamb & Kling's (2003) central thesis—the user is a social actor acting in constrained ways. The second is that systems developers who perform a good deal of the work involving the

SDM should be considered part of that artefact. This finding accords with Alter's (2003) assertion that the IT artefact can be either the technology itself, or an entire IT reliant work system. This more extensive view of the IT artefact includes task, structure, and context—where the use of the SDM is viewed as part of the work practices within an overall work system.

6 Conclusion

This study examined the relationships among systems developers, the business client, the SDM, and context that infuse meaning and purpose surrounding its use. The main contributions of the research are: the application of the social actor model outside its original domain, and a deeper understanding of the institutional forces at play in ISD. The findings demonstrate the theoretical framework (the *user as social actor model*) has been successfully applied as a conceptual lens to analyse social phenomena at the individual, project, organisational and environment levels. The findings also show how systems developers' concerns and everyday work practices are shaped by institutional structures including the SDM and the unequal authority relations that exist between the business client and systems developers. Researchers should extend these findings in several directions.

First, a natural area for future research is to study the other half in the business client-systems developer pair. While we would expect the business client to say that 'they want results and to control costs', and that 'project management is about control'; their impression of the SDM and how they view their relationship with systems developers would be insightful. Apart from Kirsch et al. (2002), there is a paucity of recent research about the management of ISD projects and the role of SDMs from the client's perspective. Second, as this paper and prior research predominantly provide insight into factors affecting SDM enactment, there is a need for process research to complement existing research in the field. What needs to be researched further is the interplay of conditions and process by which systems developers come to be involved with SDMs in their day-to-day work practices. Apart from this paper, and that of Madsen et al. (2006), there has been little attention given to the intentions, actions, context or processes surrounding SDM enactment that explain how these issues interact and how and why outcomes are associated. Process research and methods that track activities over time are needed to fill this gap.

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 & Orlikowski (1994) comment, other SDMs would mirror the same institutional contexts of systems development as described in this case

because control over 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 model, rather as discussed above the case succeeds in establishing the plausibility of the *user as social actor* model. Further research employing the sixteen characteristics and behaviours of the *user as social actor* model in other cases or a similar unit of analysis are required.

To conclude the case we claim that 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, knowledge, control, and resources that constitute the institutional context of systems development. Understanding the context and roles that development methodologies play in systems development will enable us to better understand the business client/systems developer relationship in the practice of information systems development.

Acknowledgements

This paper has benefited from the helpful comments of Gamila Shoib, Outi Brennan and the three anonymous referees. Gamila’s help in reviewing a preliminary version of the paper, and Outi’s review of *practice* in Section 4.2 is gratefully acknowledged.

References

- Alter, S., “Sidestepping the IT Artefact, Scrapping the IS Silo, and Laying Claim to Systems in Organisation”, *Communications of the Association for Information Systems*, (12), 2003, pp. 494-526.
- 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.
- Bansler, J., and Havn, E., “Improvisation in Information Systems Development”, in *Information Systems Research : Relevant Theory and Informed Practice*, B. Kaplan et al. (ed.), Proceedings of IFIP TC8/WG8.2 20th Year Retrospective, Manchester, July 15-27, 2004, Kluwer Academic Publishers Group, 631-646.

- 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.
- Bjerknes, G., and Mathiassen, L., "Improving the Customer-Supplier Relation in IT Development", Proceedings of the 33rd Hawaii International Conference on Systems Sciences (HICSS), 2000.
- Carroll, J., Howard, S., Peck, J., and Murphy, J., "A Field Study of Perceptions and Use of Mobile Telephones by 16 to 22 year Olds", *Journal of Information Technology Theory and Application*, (4:2), 2002, pp. 49-62.
- 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.
- 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., "The Use of Systems Development Methodologies in Practice: A Field Study", *Information Systems Journal*, (7:3), 1997, pp. 201-212.
- Fitzgerald, B., "An Empirical Investigation into the Adoption of Systems Development Methodologies", *Information & Management*, (34), 1998, pp. 317-328.
- Fitzgerald, B., Russo, N., and Stolterman, E., *Information Systems Development: methods in action*, McGraw Hill, Berkshire, 2002.
- Grix, J., *The Foundations of Research*, Palgrave, UK, 2004.
- Hirschheim, R., and Klein, H., "Four Paradigms of Information Systems Development", *Communications of the ACM*, (32:10), 1989, pp. 1199-1216.
- Iivari, J., Hirschheim, 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.
- Kautz, K., Hansen, B., and Jacobsen, D., "The Utilization of Information Systems Development Methodologies in Practice", *Journal of Information Technology Cases and Applications*, (6:4), 2004, pp. 1-20.
- Kautz, K., Madsen, S., and Nørbjerg, J., "Persistent Problems and Practices in Information Systems Development", *Information Systems Journal*, (17), 2007, pp. 217-239.
- Kirsch, L., Sambamurthy, V., Ko, D-G., and Purvis, R., "Controlling Information Systems Development Projects: The View from the Client", *Management Science*, (48:4), 2002, pp. 484-498.
- King, J., Gurbaxani, V., Kramer, K., McFarlane, W., Raman, K and Yap, C., "Institutional Factors in IT Innovation", *Information Systems Research*, (5:2), 1994, pp. 139-169.
- 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.

- 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 Bjørn-Anderson, N., "Power over Users: its exercise by systems professionals", *Communications of the ACM*, (30:6), 1987, pp. 498-504.
- 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.
- Sawyer, S., and Chen, T., "Conceptualising Information Technology and Studying Information Systems: Trends and Issues", in *Global and International Discourse about Information Technology*, Myers, M., Whitley, E., Wynn, E., and DeGross, J. (Eds), Boston, Kluwer, 2002, pp. 109-131.
- Schwandt, T., *Dictionary of Qualitative Inquiry*, 2nd ed. Sage, 2001.
- Scott, W., *Institutions and Organisations*, 2nd ed. Sage, 2001.
- Truex, D., Baskerville, R., and Travis, J., "Amethodical Systems Development: the deferred meaning of systems development methods", *Accounting Management & Information Technologies*, (10:1), 2000, pp. 53-79.
- Walsham, G., *Interpreting Information Systems in Organisations*, John Wiley, Chichester, 2001.
- Wynekoop, J., and Russo, N., "Systems Development Methodologies: unanswered questions", *Journal of Information Technology*, (10), 1995, pp. 65-73.
- Wynekoop, J., and Russo, N., "Studying Systems Development Methodologies: an examination of research methods", *Information Systems Journal*, (7), 1997, pp. 47-65.
- Yin, R., *Case Study Research: Design and Methods*, 2nd ed, Sage, Beverly Hills, California, 1994.

