

Effective Communication in Requirements Elicitation: A Comparison of Methodologies

Jane Coughlan and Robert D. Macredie

Department of Information Systems and Computing, Brunel University, Uxbridge, UK

The elicitation or communication of user requirements comprises an early and critical but highly error-prone stage in system development. Socially oriented methodologies provide more support for user involvement in design than the rigidity of more traditional methods, facilitating the degree of user–designer communication and the ‘capture’ of requirements. A more emergent and collaborative view of requirements elicitation and communication is required to encompass the user, contextual and organisational factors. From this accompanying literature in communication issues in requirements elicitation, a four-dimensional framework is outlined and used to appraise comparatively four different methodologies seeking to promote a closer working relationship between users and designers. The facilitation of communication between users and designers is subject to discussion of the ways in which communicative activities can be ‘optimised’ for successful requirements gathering, by making recommendations based on the four dimensions to provide fruitful considerations for system designers.

Keywords: Communication; Context; Elicitation; Requirements; Socially oriented methodologies; User–designer interaction

1. Introduction

Understanding users, their needs and how they operate within the context of the proposed system and as part of the wider organisational setting can greatly increase the likelihood of successful projects, particularly in terms of

increased real-world accuracy and completeness of the requirements document. This finding has superseded, to some degree, rationalistic perspectives on design, which have focused on the purely technical and functional requirements of a system. Indeed, in order for a system to attain a greater degree of usefulness and usability [1], there has to be an understanding of the problem area (which includes the users, the work context and the wider organisation in which the problem area is situated) and how the new technology can be applied [2]. The problem area is the ‘source’ of all requirements and a comprehensive, if ideal, understanding of this will ensure the usability of a system. The degree of understanding however, depends on the way that requirements are communicated, where user-involved approaches entail richer, if not fruitful, communicative activities for requirements gathering.

The requirements phase of a development project is characterised by intense communication activities and involves a diverse range of people who differ on levels of background, skill, knowledge and status. The goal of such activities, as already asserted, is to achieve an understanding of the problem and one that must be *shared* between disparate people, a task made all the more difficult by the complexity, vastness and volatility of the requirements. Furthermore, an increased amount of communicative effort is required to surpass the semantic gap that estranged parties, such as users and designers, inevitably foster [3]. Unsurprisingly then, effective communication has been notoriously difficult to achieve and is a recurring problem in the elicitation of requirements [4]. The overriding reason for the existence of communication problems lies in the fact that system design and development is very much a behavioural process, where human and organisational elements have an important bearing on the design. A seminal study by

Correspondence and offprint requests to: Jane Coughlan, Department of Information Systems and Computing, Brunel University, Uxbridge UB8 3PH, UK. Jane-Lisa.Coughlan@brunel.ac.uk

Curtis et al. [5] proposed such a view with overwhelming evidence from their field study, strongly supporting the notion that effective communication is crucial in system design. Despite being over 10 years old, the findings of the paper still hold true (see Al-Rawas and Easterbrook [6] for an update), even though more socio-technical methodologies have evolved in an effort to address the human element in design and attempt to curb communication problems.

There is an imperative then to analyse the complex issue of communication of requirements in the design of systems again, particularly from the user–designer perspective. With a view to this, the paper is further divided into seven additional sections. The second section discusses a socially oriented view of design in which communication is of key importance, though despite the benefits of being sensitive to communication issues, problems still abound and these are detailed in Section 3. Given these problems in communication, the review focuses on particular areas in the literature that have been pinpointed as being important in the elicitation of requirements and so a four-dimensional framework is presented in Section 4, which is used to analyse comparatively a set of socio-technical methodologies (Section 5) and their respective requirements-gathering stages, on the basis of the problems and themes drawn out from Section 4. Section 6 makes recommendations, provided from the analysis, on optimal communication activities, from a perspective that is argued might improve requirements gathering. Section 7 highlights directions where future research might prove most fruitful.

2. Design as a Social Process: The Importance of Effective Communication

There are numerous perspectives that can be taken on the design process, but the most obvious dichotomy that can be made is between the rationalistic problem-solving view, as typified by structured methodologies, versus the user-centred (problem-finding) view of the socio-technical school. The rationalistic perspective implies that there is a definable problem, solvable through a specification of the requirements and progressing in logical steps to the development of a (stable) system. Complementary to this view is one that is more socially oriented, which suggests the problem area is vague and ambiguous and needs to be located and defined in context. This viewpoint advocates an emergent and collaborative nature to requirements, suggesting that they do not simply exist in the social setting awaiting

capture, but emerge as part of ongoing interactions and negotiations between participants in the setting and outsiders such as requirements engineers.

One major and distinguishing characteristic between these two broad perspectives is the degree of communication between the designers and users. System design, especially at the requirements stage, is a social and communicative activity. However, the assumptions behind the two conflicting views are product-centred in nature on one hand and human-centred on the other and so tend to discriminate by the degree and quality of communication. Denning and Dargan [7] highlight three key assumptions of a product-centred and human-centred design process, which have been adapted into a comparison table (Table 1) to highlight relevant differences.

The assumptions behind these two divergent approaches clearly define the structure, degree and quality of communication and propose certain communicative activities and behaviours. Firstly, the goals produce differing outcomes. These outcomes are steered towards by the particular communicative activities in which users and designers engage, which are largely determined by the flexibility in behaviour that is adopted. So, the strict determinism and drive towards the creation of the end product preclude effective communication with the various parties involved and affected by the design process, largely because any stakeholder input is leapfrogged. Traditional methodologies if rigidly applied all but eliminate the ‘user’ from design, dealing with user communications from a problem-solving perspective. Such a restrictive viewpoint affects requirements capture by placing a stranglehold on communication, which is left totally unsupported in the traditional software development process [2]. However, relinquishing the problem-solving view for one that looks to problem *facilitation*, that is, spending more time communicating and developing productive relationships with users, has proven benefits as it has been linked to greater success in systems analysis and the determination of requirements [8,9].

A human-centred viewpoint proposes that the focus move away from traditional abstraction and reduction to

Table 1. Differences in assumptions on product and human centred approaches

Assumptions	Product	Human
Goals	Completed system	Customer satisfaction
Derivation of specification	Given/extracted by the customer/user	User–designer collaboration
Nature of user–designer communication	Contractual	Continual

a comprehension of the user in their environment and concrete real-life occurrences. Indeed, much of the crucial information that requirements engineers need is embedded in the social worlds of users and managers as part of the interactive and collaborative activities that occur in situ as part of the everyday [10]. This presents a dynamism and a requirements conflict, thereby making it impossible for design to be the logical outcome of the requirements [11]. Requirements in fact emerge and are negotiated and, indeed, at the heart of effective communication lies a shared understanding, which is an essential part of the successful working of multi-disciplinary design teams. This, however, can be seen as somewhat of an elusive ideal. While enhanced communication is desirable, there are many stumbling blocks to a shared understanding, which affect human-centred as well as product-centred approaches. The nature of the problems in communication are discussed next.

3. Problems in Design: Communication Difficulties

Whether it is agreed that requirements are engineered or naturally emerge through the requirements process, it must be conceded that there are a number of obstacles to realising a ‘completed’ system. These problems are not just technical in nature but also social, and it is these immeasurable factors that traditional methods try to avoid because of their unpredictability, and more socially oriented approaches attempt to address. There have been numerous studies that have categorised the different types of problems in system design, and most if not all of them are as a result of a breakdown in communication. Lyytinen and Hirschheim [12] classified system failures into four types: correspondence; process; interaction; and expectation. Excluding the first type of failure from her analysis, Macaulay [13] identified five possible causes of the three remaining types of failure, presented below in descending order of effect:

1. poor communication between people (process, interaction, expectation);
2. lack of appropriate knowledge or shared understanding (interaction, expectation);
3. inappropriate, incomplete or inaccurate documentation (interaction, expectation);
4. lack of a systematic process (process);
5. poor management of people or resources (process).

As can be seen from the analysis, it is not unreasonable to concur with Macaulay’s [13] conclusion that communication can be seen to be a key factor in the design of successful systems. However, what is also noticeable is that communication (or lack of) is an

important issue in the second and third types of causes. For example, in order to foster understanding between users and designers, there needs to be an exchange of domain knowledge. This can arise through the use of various techniques to facilitate a sharing of this knowledge and promote communication. In addition, an incomplete requirements document is a result of context avoidance and poor communication and subsequently forms an ineffective means of communication onto the next stage of design.

Other studies have narrowed down general system design failures to elicitation difficulties, which have been grouped into three problem areas: scope; understanding; and volatility [14,15]. Again, communication factors feature in all of the three problem areas, especially in ‘understanding’. Problems of understanding, particularly during the elicitation stage of the requirements process, present a major stumbling block to the success of a system because it means that ultimately the user needs will not be addressed. Furthermore, particular types of communication problems exist that can be subsumed under this area. Al-Rawas and Easterbrook [6] identified three from their field study: ineffective communication channels; restrictive notation languages; and social and organisational factors.

All of these communication problems will affect and inhibit shared understanding between users and designers. Given the multifaceted nature of communication, these problems will result in a lack of clear understanding, which will be manifest in different ways. All indicate, though some less explicitly than others, that communication has been affected detrimentally. Poor communication can be expressed in at least three different ways: articulation; misunderstanding; and conflict (e.g., see Sutton [16] for discussion on articulation and understanding and Walz et al. [17] for discussion on conflict). They can be broadly defined as follows: *articulation* – the ability to express information; *misunderstanding* – divided interpretations on the same piece of information; *conflict* – multi-perspective viewpoints and/or disagreements. These difficulties afflict both users and designers alike and occur within user and designer groups, as well as between.

These different aspects of communication are not of course mutually exclusive but influence each other and can combine to compound the effects of poor communication in certain circumstances. Indeed, the design situation is such that the problem under scrutiny is ambiguous in nature and the larger the system and wider the distribution of the design team the more the problems with communication will be exacerbated. However, despite the troublesome nature of communication problems, they can in fact provide opportunities for learning. For instance, the presence of conflict, rather

than being completely debilitating to the design process, can if channelled directly, become a mechanism for enabling learning, where the presence of a facilitator can coordinate viewpoints [17]. What is important for users and designers alike to understand, aside from the requirements themselves, is that these communication problems have stemmed from a root cause, but as requirements are enmeshed in organisational processes and come from a particular point of view [18] the cause of any difficulties is not always apparent. For instance, the presence of conflict may be a result of poor articulation of user requirements, which leads to a misunderstanding, producing an inaccurate requirements document. It could also be that political concerns are at play and may be at odds with what the design process is (rationally) trying to achieve [19].

4. Methodology and the Structure of Communication: A Framework

The management and minimisation of these communication problems require that any methodological approach to design will enable a shared understanding of the user and the context of the organisation and steps to support the user at every stage of the design process in the elicitation of requirements. In light of this, a proposal is made for a framework on four dimensions (adapted from a classification scheme by Rolland et al. [20]) with which to examine a set of methodological approaches to requirements elicitation. The choice of dimensions was based on a review of the literature pertaining to communication issues in requirements elicitation and design. They also form part of the dedicated structure of activities that are performed during the development process as part of engaging the user in the design process [21]. Therefore, this framework is advocated as a way of presenting a synthesis of the literature in these different

areas for a comprehensive discussion on a methodological approach to requirements elicitation and the support that may be provided for effective communicative practices. The dimensions under consideration are: user participation and selection; user–designer interaction; communication activities; techniques and representation (though this last dimension is not included in the comparative analysis or included in the diagram below; see Section 4.5 for more explanation). The framework is presented in Fig. 1.

4.1. User Participation and Selection

A representative audience is a vital prerequisite for communication activities in requirements gathering and will also have a strong bearing on the other dimensions of design, by the mere fact that communication can be rendered ineffective by the presence of inappropriate people whose contribution will be limited. Therefore, user participation and selection can be defined as the early focus on and selection of appropriate candidates, who are expected to participate in an active and direct manner (ideally) so as to establish workable communication links. However, issues concerning the user in the requirements' stage form a common cluster of problems (see Woolgar [22] for a small sample of a vastly extendable list). The main problems lie in identifying the 'user' and the difficulty the user has in articulating requirements, that is, knowledge of their work relevant to the design of a system. This knowledge can be split along two major dimensions: tacit and explicit (following Polanyi [23]), or even illicit in the sense of the information held by users being in fact invalid.

Obviously, in terms of the communication of requirements, explicit knowledge can be reasonably expressed. However, it is the tacit knowledge held by the user which is of great value to designers, as it arises from the user's involvement with the world [24]. Knowing more than you can tell [23] makes this knowledge the hardest to elicit, especially in the traditional question and answer type process of extraction and if users are expected to discuss what they know in the context of design problems [25]. While it is beyond the scope of this paper to enter into a detailed discussion on the nature of knowledge, especially tacit knowledge (but see Stenmark [26]), it is sufficient to say that tacit knowledge concerns the execution of action that is highly situated in nature [27]. The problem of eliciting tacit knowledge is two-pronged [16]. (1) Users have difficulty understanding their own requirements, that is, knowing what is actually involved in a performance of a task and then moving from this reflective state to a

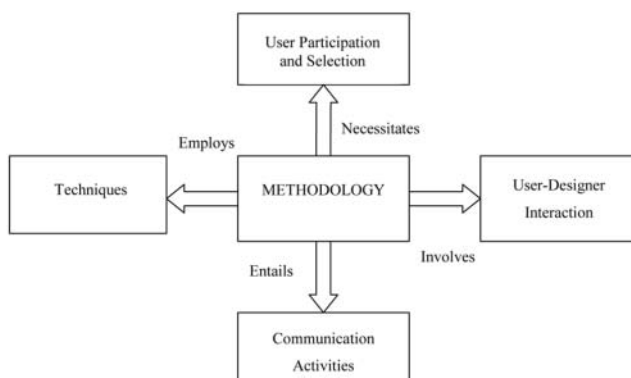


Fig. 1. Four-dimensional view on a methodology (adapted from Rolland et al. [20]).

statement of what it is that they may want. (2) Users' explanations of what they do in a task will not be made as an explicit statement of requirements but framed more personally and therefore ambiguously. The challenge for requirements engineers is trying to bridge this gap between the tacit and the explicit in order to elicit requirements that will be suitable for the design of systems. Maintaining the context for the user, in the way that a task may be performed, is important in teasing out the tacit forms of action. Specific techniques for elicitation have been favoured such as prototyping, where having the product on hand can enable communication in such a way as to enhance the cooperative nature of working between users and designers and provide a form of representation that may render the user's work more visible [28].

At a bare minimum, brief consultation of users, or even one or two user representatives in a design team, could constitute an enlistment of user participation. However, there are many different needs and interests

that need accommodation and representation in a design team and, therefore, various ways of identifying suitable user participants (e.g., [29,30]). A hypothetical group of participating users could comprise of members with one or more of the following attributes defined below. These attributes can be used as classifications of different stakeholder types and represent facets of the dimension of user participation and selection as shown in Table 2.

1. *Specific task knowledge and skill* – this should include users, from frequent users to those who might be affected by the system, so as to obtain the widest spread of domain knowledge.
2. *Status* – this refers to managers of high–low status who will share responsibility in the human sense of ensuring implementation and acceptance of the system.
3. *Responsibility* – this refers to responsibility in the technical sense, design and development of the system and calculating the finance available in the budget.

Table 2. A comparison of approaches with respect to dimensions and facets relevant to communication (drawn from Section 3)

Dimensions	Facets	Selected approaches			
		MUST	JAD	ULRC	SSM
User participation and selection	<i>Stakeholder type</i>				
	Specific task knowledge and skill	Future users (Design team)	Users who will use or be affected by the system	Prospective users	Appropriate subset of users
	Status	Management (Steering committee)	Management	Management input unclear	Management
User–designer interaction	Responsibility	IT professionals (Design team)	IS staff	Users (IT staff involved in initial training)	IT staff involved in later stages
	<i>Roles</i>				
	User role	(towards) Expert	Co-designer	Designer/Model builder	(towards) Co-designer
Communication activities	Designer role	(towards) Apprentice	Facilitator	Trainer	(towards) Expert
	<i>Behaviours</i>				
	Knowledge acquisition	Focus on user and context of system use (social work practice and political fit)	Focus on the JAD process	Users take responsibility for modelling requirements	Focus on user and context of system use (social work practice and political fit)
Techniques	Knowledge negotiating	Sharing of perspectives between users and designers	Limited view of user perspective from designers	Limited views of both user and designer perspectives	Sharing of perspectives between users and designers
	User acceptance	Cooperative	Cooperative	Semi-cooperative	Cooperative
	<i>Class</i>				
	Traditional	✓	X	✓	✓
	Group	✓	✓	✓	✓
	Prototyping	✓	X	X	X
	Model-driven	✓	X	✓	✓
	Cognitive	✓	X	X	X
	Contextual	✓	X	✓	✓

✓ indicates use of particular class of technique; X indicates class of technique is not used.

Intuitively, user/stakeholder selection should be on the basis of domain knowledge, given that this is precisely the type of knowledge that requirements engineers lack. For example, Lubars et al. [31] concluded from interviews with software developers that a high level of domain expertise is an important factor in determining the success of requirements engineering (RE), because it helped to avoid misunderstandings between the RE team and other stakeholders, though stakeholder contact was fairly minimal overall in the projects under study. Other studies have shown that user representatives tend to be chosen on the basis of their position and status rather than their knowledge per se [32]. Project constraints will dictate ultimately who should be involved, but the motivating force behind user selection and participation should be that the best people be chosen for the design team, or that the best be made out of the chosen few.

After satisfying the selection criteria, user participation needs to be organised in such a way that users receive the support and guidance that they need on the different roles they are meant to play in the requirements process and beyond. It is inevitable that as part of a team structure members will adopt roles and relationships relative to each other [33]. Given the difficulties posed by a heterogeneous design team, the adoption of roles holds implications for communication as mediating posts, as it were, for the transfer of design requirements. For example, Sonnenwald [34] has identified different types of communication roles across four different multidisciplinary design situations. The uptake of these roles provides the means through which knowledge exploration, collaboration and negotiation, in the case of conflict, can be enabled, all of these activities being important in the successful communication of design requirements (see Section 4.3).

4.2. User-Designer Interaction

Particular role adoption can influence the quality of the interaction and taking it as the smallest unit of communication analysis, this is used to define user-designer interaction. Roles are classified as either being user or designer held and represent facets of the dimension of user-designer interaction shown in Table 2. This relationship has been portrayed in the literature as being highly problematic in nature and characterised by a culture gap. Designers tend to lack domain or business knowledge and consequently tend to misunderstand or ignore some requirements and their social context. Socially oriented approaches to elicitation force a collaborative role-play that elevates the user to an equal footing with the designer so that joint decision-making is possible as well as the satisfaction of both

parties. For example, Boland [35] compared two different structures of interaction between a designer and a manager – problem-solving (designer as problem-solver) and problem-finding (both parties develop a mutual partnership in dealing with problems) – and found that the latter type of interaction process produces higher-quality designs along with important implementation advantages. The stance that the designer takes in the requirements elicitation process determines the way that information is exchanged and a problem-finding mode of enquiry will foster the development of different viewpoints and thus a shared understanding of the problem and potential solutions.

An extremely useful model of user-designer communication, which expands on this idea of problem-finding and interaction, is by Kensing and Munk-Madsen [36]. This stands in opposition to traditional models, generally founded on the concept of ‘transmission’, whereby the interaction between user and designer seems to exist only as a way of obtaining information that is to be passed on to the next stage of design. Alternatively, their model is based far more on the concept of interaction, and involves three domains of discourse in design (the users’ present work, the technological options and the new system) on two levels of abstraction (abstract and concrete), resulting in six areas of knowledge. These areas of knowledge require integration for the design process to be a success and the use of particular techniques will influence the type of knowledge developed (see Section 4.4).

In contrast, Beyer and Holtzblatt [37] regard the problem in the relationship between users and designers as one of enabling learning. Hence the model they propose is one of the designer-as-apprentice, whereby the users become the experts in their work and the designers can learn from the user’s (or the master’s) experience. Obviously, designers have greater responsibilities than a mere apprentice, in that ultimately they are replete with skills necessary for the production of a system specification. However, casting the relationship in this way fosters a greater degree of understanding and connection with the user and therefore offers a prime opportunity in which to explore the work of the user and its potential relevance to the design of a new system.

A common thread between these two models is that a successful user-designer relationship necessitates cooperation in order to share knowledge or learning of importance for requirements gathering. The nature of cooperation is determined by the different roles, not necessarily static, that are adopted by users and designers. It is a difficult role-play since users and designers are both experts in their respective fields and nothing short of a collaborative enterprise will succeed, especially in the construction of requirements. Commu-

nication in the requirements arena means that developing a shared understanding of an ambiguous situation is of utmost importance. The root of the requirements problems lies in the common ground between the user and designer, which can only be discovered through communication activities that facilitate a sharing of information.

4.3. Communication Activities

The activities that are undertaken by the design team as part of a methodological framework also have a bearing on the degree of communication between users and designers. Indeed, Hartwick and Barki [21] identify ‘communication activity’ as an important dimension of user participation in the development process. They define it as activities involving formal or informal information exchange between all stakeholders in a project. This definition is deemed accurate for the dimension of communication activity as discussed in this paper, though in terms of requirements’ elicitation any activities are communication activities, but they should be structured in a way that promotes effective communication. For communication to occur reliably in the realm of requirements, there needs to be a shared understanding, which can only occur through cooperation and negotiation. The basic and most productive behaviours of a communication activity programme revolve around knowledge acquisition, sharing and integration activities [17] and the coordination of the efforts involved [38] – all of which can be used to address particular areas that have been identified in Kensing and Munk-Madsen’s [36] model, outlined in Section 4.2.

The main behaviours as part of a communication activity, then, are (based on Walz et al. [17]):

1. *Knowledge acquisition* – There are links that need to be made between the user’s and designer’s realms of knowledge and experience and of the technological options, so as to achieve a shared understanding and common vision of a future system. In its most encompassing form this involves focus on the user and context of system of use and consideration of the social, work practice and political fit.
2. *Knowledge negotiating (sharing)* – Requirements need to be negotiated as part of an iterative process, which helps to define the requirements through a thorough understanding of each other’s (user and designer) perspective.
3. *User acceptance* – Acceptance of the system implies integration of the user–designer viewpoints where both parties cooperate to understand the scope of the

system and are satisfied that it will work within the limitations imposed (e.g., particular work relations, organisational structure).

Communication activities can be classified according to these behaviours and represent facets of the communication activity dimension as shown in Table 2. These activities provide a structure but they cannot be considered without relation to the techniques that can act as mediators for communication. Of note here is the idea of mediation in conjunction with artefacts, which is a key feature in other theoretical frameworks on the study of context and human practices contained therein, such as activity theory. However, activity theory details a specific notion of context and goal-oriented interaction that is beyond the scope of this paper, although its findings provide another perspective on communication studies of this kind [39,40].

4.4. Techniques

The techniques used in the design process are of great importance at the requirements elicitation stage as they provide a useful means of facilitating communication. For example, these techniques have been defined as customer–developer links [41], which allow an exchange of information. The authors further distinguish between direct and indirect links where, from a communication perspective, direct links (and an increased number) are preferable in providing face-to-face contact and a multiplicity of cues to enrich communication and thus reduce ambiguity, which is common in a requirements capture situation.

In essence, an elicitation technique can be defined as a method for mediating communication (though more indirectly in the case of contextual techniques) and, for brevity, six broad classes of elicitation techniques for the requirements of systems can be identified (following Nuseibeh and Easterbrook [42]) and are shown in Table 2 and below:

1. *traditional* (e.g., questionnaires, interviews, analysis of existing documentation);
2. *group* (e.g., brainstorming, focus groups, consensus-building workshops);
3. *prototyping* (e.g., mock-ups);
4. *model-driven* (e.g., scenarios, rich pictures);
5. *cognitive* (e.g., protocol analysis);
6. *contextual* (e.g., ethnography).¹

Evidently, some of these techniques are more effective in the elicitation of requirements than others, depending

¹MUST employs ethnographic techniques as a specific step in the methodology, while ULRC and SSM employ ethnographic techniques if deemed necessary to gain more information.

on the context under study. In addition, different techniques are useful for eliciting particular kinds of requirements. For instance, contextual techniques are more likely to uncover tacit areas of knowledge than are questionnaires [43]. Furthermore, there is the issue of support. In the requirements process, users need support through the difficult process of negotiating the requirements, and particular techniques afford such support. For instance, group session techniques are more amenable to the active encouragement and exchange of ideas, whereas traditional techniques do not support the user in the requirements process or inject any concrete meaning into the process.

4.5. Representation

A perhaps obvious point of communication breakdown is in the specification document, which in essence is a medium for the communication of requirements. However, the design team tends to be composed of a diverse range of people and the language they use in terms of system specification will be different. The naïve user will prefer natural language but designers will often use notations that are unfamiliar to the user, which causes problems at the validation stage [6]. A thorough discussion of the problems in representation is beyond the scope of this paper, but it is important to note that a successful communication process can be hindered at the representation stage if users cannot relate to the requirements being validated back to them through the specification document. Further, this aspect of design, although important in communicative terms, is not considered in the methodological comparison as the argument is primarily based around the critical factors involved in the elicitation of requirements. So it is within this loose boundary of communication ‘ice-breaking’ that the discussion in the remainder of this paper is confined and the comparative analysis follows.

5. Requirements Elicitation: Types of Methodologies

Having discussed aspects of systems design that are pivotal to requirements elicitation and underpinned by communication activities, this section focuses on the types of methodology in which these aspects tend to be well represented, that is, those that aim to capture a broader view of system requirements than just the technical; these are the social, organisational and human aspects of design. These socio-technical methodologies

inevitably entail a high degree of user involvement and communication links as part of an iterative approach to design, though in different ways.

Previous work has attempted comparisons between different methodologies on the basis of user involvement. Carmel et al. [44] make a straightforward comparison of participatory design (PD) and joint application design (JAD); Beckworth and Garner [15] attempt a more in-depth analysis by comparing different requirements engineering methods against perceived problems of scoping, understanding and volatility and their associated issues; and Bekker and Long [45] examine the similarities and differences of a selection of user-involved design approaches and produce, what could appear, at least to the uninitiated, a bewildering though useful list of methodological attributes, divided into non-configurable and configurable.

This paper, however, attempts to organise the analysis in a more thematic way, and from the perspective of communication, by examining the dimensions critical in ensuring effective communication in requirements elicitation and in a way that is particularly pertinent to the requirements engineering literature. The comparative analysis considers four such methodologies and how they are structured, briefly introduced below in Sections 5.1–5.4. The justification for the selection of the methodologies can be found in Section 5.5.

5.1. MUST²

MUST represents a coherent method for participatory design [46]. It is grounded in six principles (following in the vein of the participatory design tradition) and offers a set of techniques for representing current work and future computer-based systems. The overall design process itself is constituted by five main activities. Project establishment (analysis of the problem) and strategic analysis (clarification of appropriate work domains to focus on) are two closely linked activities. In-depth analysis of selected work domains is the third main activity for the understanding of current work practice rationales. Developing visions for overall change is the fourth but most central activity, conducted primarily through workshops. Finally, anchoring the visions is an activity that roots the ‘vision’ in the organisation, so that is owned by the users; this is only achieved through such a strongly participatory approach.

²MUST is a Danish acronym for theories of and methods for design activities.

5.2. Joint Application Design (JAD)

The main impetus for the development of JAD by IBM in the late 1970s was to set up structured meetings, known as the JAD session, to improve communication between user representatives and expedite decision-making with all present [47]. JAD consists of five phases. The first two phases, project definition and research, comprise various elicitation tasks, information from which is used as part of the third phase, preparation for the JAD session. Indeed, the whole JAD process is centred on this group session (phase four), which is run by a facilitator to coordinate the group's contributions and validates the information that has already been gathered to produce the final document in phase five.

5.3. User-Led Requirements Construction (ULRC)

ULRC is essentially a social process that addresses a major problem in the requirements elicitation process, that of the user-developer culture gap [48]. This can be overcome by training users to build the requirements models themselves. The approach comprises three rounds. Round one consists of user training, where the developer transfers knowledge and skills to the user for the building of the requirements model. Round two consists of constructing a model of the current domain, used as the basis for constructing a model of the future domain in round three as part of an iterative process to improve the quality of requirements.

5.4. Soft Systems Methodology (SSM)

As a methodology, SSM offers a set of guidelines, which can be applied to 'messy, changing, ill-defined situations' [49]. The core idea is that people work through seven phases of the methodology (though not necessarily linearly) in order to analyse complex systems to plan and determine appropriate changes. In essence, the seven-phased process is very simple. It starts with a formulation of the problem, represented in pictorial form (phases one to three). This 'rich picture' allows for the elicitation and development of multiple perspectives of the system, one new perspective of which is taken to develop a conceptual model (phase four) to contrast against the existing problem situation (phase five). The comparison of desired and actual states is debated by participants (phase six), in order to refine the requirements and take action to improve the problem situation.

5.5. Selection of Methodologies

The methodologies have been chosen in order to highlight differences between implicit communication structures (they are also contemporary, and therefore represent approaches that are seen as currently relevant in research terms). A conceptualisation of the methodologies from a communications perspective is presented in Fig. 2. The methodologies have been classified along two different dimensions: control and scope. Bekker and Long [45] identified these two attributes as part of a comparative analysis of user-involved approaches, labelled there as 'user control' and 'design problem scope', where user control was described as either weak or strong and design problem scope focused on either task, social or political fit.

The control of communication is important to take into account as it directs the entire requirements elicitation process. In this selection of methodologies, communicative control and responsibility can be said to either reside with the user (e.g., ULRC and MUST, though it should be noted that the latter is more of a combined effort between user and designer), or with the designer/facilitator (e.g., JAD and SSM). The scope of the ensuing elicitation activities is another important factor for evaluation as it determines whether the design problem is viewed as extending beyond technological considerations of task, referred to in Fig. 2 as 'system in use' (also in recognition that some more user-involved approaches can be product-centred in nature in their adherence to a more completed systems view of requirements). Alternatively, the scope can encompass more human-centred considerations, such as social and political contextual factors. Here the direction of efforts is on the 'use of the system', where the drive is towards a more balanced view of social and technical considerations as to how the system will be used and in the context in which it is to be placed.

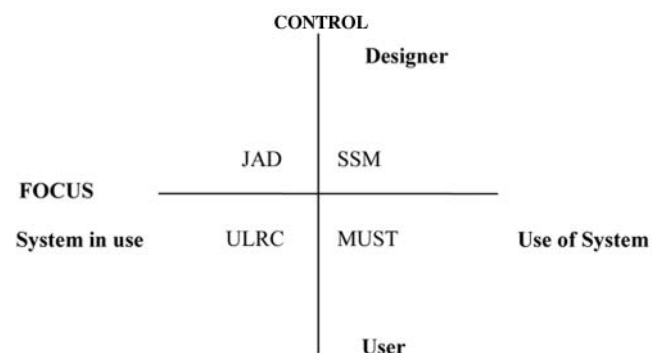


Fig. 2. A classification framework for the methodologies under analysis.

5.6. Comparative Analysis

Up to this point, the paper has argued for a more emergent and collaborative view to requirements elicitation and communication, which serves to build user–designer relations and allows for a sensitivity to user needs, contextual and organisational factors, all of which are important in the successful design of systems. The paper has focused on four different dimensions that have been viewed as crucial in the communication of requirements and are therefore earmarked as important structures in any methodological approach to requirements elicitation. The paper now turns to a comparative analysis of four different methodologies using the different dimensions discussed in Section 4 of this paper, with the analysis summarised in Table 2. The details relevant to communication discussed within the key dimensions of design were deduced from the literature (see Sections 4.1–4.4) as being critical factors in any methodological approach to the requirements process. A full discussion of the comparative analysis, as represented in Table 2, follows in Sections 5.7–5.10.

5.7. User Participation and Selection

While user involvement is a laudable concept, it is also vital that the *right* people are involved to ensure effective communication. A list of possible stakeholders is often suggested, based on skill, status and responsibility, to impart the fact that users can offer a variable degree of input in the design process that must be accommodated in some way. A truly representative group of stakeholders is notoriously difficult to achieve, given the fact that other influencing factors, such as constraints on an individual's time (also dictated by the organisation), may exclude that person from participating.

None of the approaches make a clear statement of stakeholder participation, or indicate the selection criteria of individuals involved. MUST does involve efforts to include all appropriate stakeholders and also provides for a division of teams, namely a design team (IT professionals and future users) and a steering committee (managers – organisational and IT – and one/two user representatives). The rationale behind the first team is to design and the second team is to make decisions so that the project can progress, a set-up that appears to work in practice [50]. Reasons for this success is that the steering committee acts as something of a facilitator in the mediation of conflict and decision-making.

JAD also endeavours to select the 'right' people, necessitated by the rigid team structure to its approach

and its commitment to fast development times. Again, the detail of the selection process is not made explicit despite the fact it is often put forward as a method of enhancing user participation. Similarly, ULRC and SSM also suffer from a lack of direction on stakeholder involvement. SSM, however, has a good 30-year long history of use which is relatively supportive of its practice [51], while ULRC, being newly developed, has no real evidence of substance to attest to its effectiveness in practice.

5.8. User–Designer Interaction

The four different methodologies present various interaction models between users and designers. The question is whether there is an optimal level of interaction in order to communicate effectively. This must be dependent on the proposed system and context in question. It is clear from Table 2 that with MUST the focus is more on the working relationships between users and designers, which is continually reinforced throughout the methodological process; this is equally the case with SSM. In contrast, JAD's structure forces a rigid uptake of roles, with the facilitator's role being all-important in maintaining the drive towards the creation of the end product. With regard to ULRC, the interaction is very different as design is completely user-led, that is, communicative responsibility lies with the user. An evaluation study of the training process, undertaken by the users and conducted by Flynn and Jazi [48], suggests some benefits to this approach. One comment from a user on the communicative value of the methodology was regarding the diagrams produced by the modelling process. They were found to be confusing at first, though easy to read after persistent study. However, more independent research is required to validate these claims and with larger groups of users.

5.9. Communication Activities

MUST is the most intensely communicative methodology of the four as every stage involves close collaboration between users and IT professionals. It is strong on knowledge acquisition and negotiating, which naturally leads to the user accepting the system, as they have been so closely involved in the process. SSM encourages alternative views of the situation, which are formalised in a root definition, so that the comparison between current and future uses of the system is continual. JAD is weak at acquiring knowledge, allowing the design meetings to provide the opportunities for this type of information to arise. It is extremely structured – so much so that communication activities

are highly directed, especially given the presence of a facilitator. This downgrades any negotiation, and while the user may accept the system it is not necessarily with the same sense of ownership as with users involved in MUST. In ULRC, the training concerns communicative activities in modelling the organisation and conveying this information to others. The activities involve all of the three types of communication behaviours, though the success of the activities is debatable.

5.10. Techniques

Of all the methodologies, MUST is most flexible in employing a wide variety of techniques, which create communication links and a close study of the context under examination. Wood and Silver [47], in describing JAD, said ‘first the methodology, then the tools’ (p. 179). So while certain techniques may be useful in supporting the JAD process, it is intensely group-focused and much responsibility is placed on the skill of the facilitator to direct the session, which must remain focused so as not to wander away from the topic of requirements [3]. In ULRC, the techniques are used merely as a vehicle to empower the designer to be equipped to train and the user to be able to carry out the modelling tasks. Their skill in this then becomes of communicative significance as it enables the dissemination of information so that the system can be built. SSM also employs a wide variety of techniques in order to develop a rich picture in which to promote further communications.

6. Conclusions

This comparative analysis has attempted to offer meaningful insights into the communication practices of the system design process at the requirements level. An understanding of the way in which communication is facilitated within the design process has implications for the type of methodology adopted as research has shown that problems in communication have a direct bearing on project outcomes. The conclusions that can be drawn from the preceding analysis are divided into two sections for clarity. Section 6.1 first discusses the practicalities of adopting a prescribed methodology, so as to ground the conclusions from the analysis in some real-life context of research. The conclusions in Section 6.2 end with a set of recommendations for the effective communication of requirements which can be made with adherence to the four-dimensional framework used in the analysis.

6.1. Methodology: Theory versus Practice

One drawback of this analysis is that it has been theoretical in nature and so the consequences in practical terms are limited. However, the analysis has raised some important points, in relation to the support a methodology provides in the communication of requirements. These points will require further empirical investigation, as it is often the case that the value that a methodology claims to impart in theory rarely crystallises in practice. There is little in supporting evidence positively evaluating the MUST and ULRC methodologies with respect to claims made, though these are fairly new in their development. However, where evidence does exist, it does tend to suggest, especially with JAD for example, that the methodology is fairly complicated to utilise especially if followed to the letter. MUST appears to have been adopted solely by its developers, which might betray a necessity to be familiar with its guiding principles. ULRC is a methodology still in its infancy and has no clear independent support for its successful use in practice and in a wide variety of contexts, which is in contrast to SSM’s (and JAD’s) 30-year long history of popular use.

It is not the purpose of the paper at this stage to provide an in-depth review of the evaluative research on these methodologies as used in practice. However, some evidence from two different studies on JAD in practice is mentioned here as a clear illustration of how theoretical claims, in JAD’s case of improving communication between user representatives to expedite decision-making, are not necessarily upheld, to the detriment of communication. For example, a number of findings from Davidson [52] showed that in a third of JAD projects studied workshops were attended by IS staff instead of actual business users, making it impossible to lead to any sense of relationship building. Furthermore, in another 60% of projects, the aim of design meetings to produce analytical models came at the expense of reaching consensus, improved quality and efficiency of the requirements definition. A second study by Purvis and Sambamurthy [53] detailed some user experiences with JAD and found that they were rather sceptical of the methodology. Reasons for this distrust were attributed to the user perception that designers were not particularly confident at using the methodology, especially at handling the difficult role of facilitator, causing users to lose faith in the methodology as a way of enhancing user–designer interaction. Yet more of their documented experiences with JAD suggest that there may be problems with this approach, since users do not warm to the prospect of taking responsibility for communication and negotiation of requirements [53].

The evidence then that is presented from JAD in practice, above, does at first glance cast a rather dim view of its success in organisations and a strong lack of support for the dimensions identified as important in communication (e.g., lack of adequate user representation in sessions). However, to debate the merits or otherwise of the use of any particular methodology in requirements gathering and elicitation would be making a point too wide of the mark. The fact is that an obvious paradox exists between the development of complete methodologies and their full realisation in practice [54,55]. When methodologies are used, it is more the case that parts of them are used (or parts from different methodologies) rather than following all the steps required by a particular methodology [56]. Studies with software designers, (e.g., [18,57]) highlight this apparent gap between the prescription of a methodology and the specifics of the working circumstances of a development project. Both the survey of the former and the ethnomethodological study of the latter jointly corroborated the fact that making the methodology work involves behaviour that is more ad hoc than procedural.

Therefore, using a particular methodology does not directly map onto an understanding of the organisation, context and users. It is through an understanding of the situation and the tacit knowledge that appropriate methods can be employed and put to effective use [54]. But in order for understanding to be achieved and shared, there must be effective communication. Nandhakumar and Avison [55] found in their field study that the development process was characterised by such things as improvisation, opportunism, interruption and mutual negotiation. For these contingencies to be managed effectively, support for communication is key. This requires a consolidation of the knowledge in the four-dimensional view on system design which has been presented above, where it is clear that increased and effective communication in these four areas has a direct impact on the acceptance and success of a system.

The analysis clearly identifies and puts forward four dimensions of design that should ideally be taken into account in the execution of an existing methodology or development of a new one. The methodologies under examination show differences in their approach to dealing with these four areas and encounter varying degrees of success. This success, however, is tempered by the absolute nature of the organisation itself and the limitations it will impose, so if a methodology is not organisationally focused, then however attractive it looks in theory, any benefits will be ultimately lost in practice. If a methodological approach is sensitised to the issues that an organisation presents, then communication between the IT staff and users will proceed

more efficiently. Again, this is of course context-dependent and a thorough understanding of the business is required in order to adapt any methodology to the business and user needs.

6.2. Recommendations for the Effective Communication of Requirements

The discussion reveals so far that the four-dimensional framework raises some important issues and questions with regard to the adoption of a methodology and its support for the communication of requirements. Consideration of these dimensions and associated concerns can hold value for an organisation and its stakeholders attempting to integrate technology into their context of use. Thus four clear recommendations can be made which summarise succinctly the findings from the analysis:

- The literature has demonstrated time and time again the perils of disregarding the users in design. This impedes communication with those individuals who will ultimately use the system and their exclusion will quite obviously prevent the communication of valuable requirements. Yet there still seems to be a lack of focus in requirements methods on the detailed make-up of stakeholder groups.
- Once an adequate mix of IT and business users have been selected then interaction should proceed on a cooperative basis. Both broad parties should have equal input into the design of the system so as to foster a sense of ownership and keep abreast of changes that take place in the system. Managing the level of responsibility, degree of decision-making and the balance between users and designers seems a key area where little empirical research exists to inform practice.
- Within the engagement of communication activities there appears to exist a tripartite division between knowledge acquisition, knowledge negotiation and user acceptance, though in real terms these behaviours are intimately linked. Knowledge must be acquired with sensitivity to the context, which puts the design team in a good position to negotiate the requirements with as much knowledge to hand as possible. User acceptance is a behaviour that involves the integration of viewpoints of both IT and users. Time must be spent at the front end of design to help all parties appreciate the context of the desired system, not only from a use perspective but also in terms of social and wider organisational constraints. While not immediately productive, such activities ultimately frame the likely acceptance of a system.
- All of the methodologies employ a variety of

techniques to produce the communication behaviours – apart from JAD, which has received the most criticism of its efforts in the literature. It may suggest that multiple technique use offers a way of representing the same information in different ways for validation or finding out particular types of information, such as the nuances of work practice, though more work is required on technique selection, use and derived benefits within such methodologies.

7. Directions for Future Research

The results of the analysis and discussion spotlight many areas for additional research, which when taken into account can provide a provisional agenda for future communication-centred studies both for researchers and practitioners. These are summarised below:

- Methodological structures (despite not being followed religiously) do belie a set of assumptions that are prevalent in design and this can either constrain or enhance communication, which the influence of the context or organisation can further exaggerate. More research therefore is required on methods in use that conduct studies in real-life settings employing more naturalistic techniques so as to reveal the facets of communication in action and context. This would be extremely useful to both researchers and practitioners alike in understanding how any ‘limitations’ in methodologies are overcome in order to achieve the desired objective. A question highlighted by Introna and Whitley [54] of high relevance here: What is the nature of best practice in systems development?
- The paper continually reasserts the importance of communication for practitioners and the need to build relationships with the user for effective communication. This paper provides more of a theoretical look at communication; therefore more real-world research is required in order to analyse the user–analyst relationship, in particular the form that user–designer interaction takes, whether cooperative, interrogative etc., in order to assess more closely the implications of particular styles of interaction and the context in which they are embedded so as to build up a useful set of practical guidelines that can be incorporated into a methodological approach or employed alongside.
- The four-dimensional framework as presented in the paper requires more detailed investigation and empirical evidence to check the influences they exert in real-life projects. It is also clear that there are different facets to each of these dimensions (as described in Table 2; e.g., ‘user role’ in the user–designer interaction dimension), and that these give

rise to such issues as control and responsibility of communication between user and designer. More research is needed to look more closely at these issues in order to determine and understand the nature of the behaviours that are engaged in as part of a communication activity programme or methodology for the elicitation of requirements and beyond.

In essence, the adoption of a more emergent and collaborative view to requirements elicitation engenders increased contact and involvement with users. This ultimately leads to the building of relationships, which provide a platform for the reduction and negotiation of problems in communication. The communication perspective provides a more holistic approach to the elicitation of requirements in encompassing a wide range of factors that should be taken into account for a successful requirements phase and system design.

References

1. Landauer TK. The trouble with computers: usefulness, usability and productivity. MIT Press, Boston, 1995
2. Atwood ME, Burns B, Gairing D, Girgensohn A. Facilitating communication in software development. In: Symposium on designing interactive systems, 1995, pp 65–73
3. Bostrum RP. Successful application of communication techniques to improve the systems development process. *Inform Manage* 1989;16:279–295
4. Saiedian H, Dale R. Requirements engineering: making the connection between the software developer and the customer. *Inform Software Tech* 2000;42(6):419–428
5. Curtis B, Krasner H, Iscoe N. A field study of the software design process for large systems. *Commun ACM* 1988;31(11):1268–1286
6. Al-Rawas A, Easterbrook S. Communication problems in requirements engineering: a field study. In: Proceedings of the first Westminster conference on professional awareness in software engineering, 1996
7. Denning P, Dargan P. Action-centred research. In: Winograd T, Bennett J, De Young L, Hartfield B (eds). *Bringing design to software*. Addison-Wesley, Harlow, UK, 1996, pp 105–120
8. Vitalari NP, Dickson GW. Problem solving for effective systems analysis: an experimental exploration. *Commun ACM* 1983;26(11):948–956
9. Marakas GM, Elam JJ. Semantic structuring in analyst and representation of facts in requirements analysis. *Inform Syst Res* 1998;9(1):37–63
10. Goguen JA. Requirements engineering as the reconciliation of social and technical issues. In: Jirotko M, Goguen J (eds). *Requirements engineering: social and technical issues*. Academic Press, London, 1994, pp 165–199
11. Pye D. The nature and aesthetics of design. Van Nostrand Reinhold, New York, 1978
12. Lyytinen K, Hirschheim R. Information system failures: a survey and classification of the empirical literature. *Oxford Surveys Inform Tech* 1987;4:257–309
13. Macaulay LA. *Requirements engineering*. Springer, London, 1996
14. Christel MG, Kang KC. Issues in requirements elicitation. Technical report CMU/SEI-92-TR-12 ESC-TR-92-012, Pittsburgh, PA: Carnegie Mellon University, September, 1992

15. Beckworth G, Garner B. An analysis of requirements engineering methods. Technical report TRC9524, Deakin University, School of Computing and Mathematics, 1994
16. Sutton DC. Linguistic problems with requirements and knowledge elicitation. *Req Eng* 2000;5(2):114–124
17. Walz DB, Elam JJ, Curtis B. Inside a software design team: knowledge acquisition, sharing, and integration. *Commun ACM* 1993;36(10):63–77
18. Button G, Sharrock W. Occasional practices in the work of software engineers. In: Jirotko M, Goguen J (eds). *Requirements engineering: social and technical issues*. Academic Press, London, 1994 pp 217–240
19. Franz CR, Robey D. An investigation of user-led system design: rational and political perspectives. *Commun ACM* 1984;27(12):1202–1209
20. Rolland C, Ben Achour C, Cauvet C, Ralyte J, Sutcliffe A, Maiden NAM, Jarke M, Haumer P, Pohl K, Dubois E, Heymans P. A proposal for a scenario classification framework. *Req Eng* 1998;3(1):23–47
21. Hartwick J, Barki H. Communication as a dimension of user participation. *IEEE Trans Prof Commun* 2001;44(1):21–36
22. Woolgar S. Rethinking requirements analysis: some implications of recent research into producer–consumer relationships in IT development. In: Jirotko M, Goguen J (eds). *Requirements engineering: social and technical issues*. Academic Press, London, 1994, pp 201–216
23. Polanyi M. *The tacit dimension*. Routledge & Kegan Paul, London, 1966
24. Introna LD. *Management, information and power*. Macmillan, Basingstoke, 1997
25. Randall D, Hughes J, Shapiro D. Steps toward a partnership: ethnography and system design. In: Jirotko M, Goguen J (eds). *Requirements engineering: social and technical issues*. Academic Press, London, 1994 pp 241–258
26. Stenmark, D. Turning tacit knowledge tangible. In: *Proceedings of the 33rd Hawaii international conference on system sciences*, 2000
27. Suchman L. *Plans and situated actions: the problem of human–machine communication*. Cambridge University Press, Cambridge, 1987
28. Suchman L. Making work visible. *Commun ACM* 1995;38(9):56–64
29. Macaulay L. Requirements capture as a cooperative activity. In: *Proceedings of the IEEE international symposium on requirements engineering*, 1993, pp 174–181
30. Damodaran L. User involvement in the system design process: a practical guide for users. *Behav Inform Tech* 1996;15(6):363–377
31. Lubars M, Potts C, Richter CH. A review of the state of the practice in requirements modelling. In: *Proceedings of the IEEE international symposium on requirements engineering*, 1993, pp 2–14
32. Beirne M, Ramsay H, Panteli A. Participating informally: opportunities and dilemmas in user-driven design. *Behav Inform Tech* 1998;17(5):301–310
33. Cross N, Clayburn Cross A. Observations of teamwork and social processes in design. In: Cross N, Christiaans H, Dorst K (eds). *Analysing design activity*. Wiley, Chichester, 1996 pp 291–317
34. Sonnenwald DH. Communication roles that support collaboration during the design process. *Design Studies* 1996;17(3):277–301
35. Boland RJ. The process and product of system design. *Manage Sci* 1978;24(9):887–898
36. Kensing F, Munk-Madsen A. PD: structure in the toolbox. *Commun ACM* 1993;36(4):78–85
37. Beyer HR, Holtzblatt K. Apprenticing with the customer. *Commun ACM* 1995;38(5):45–52
38. Kraut RF, Streeter LA. Coordination in software development. *Commun ACM* 1995;38(3):69–81
39. Engeström Y. *Learning by expanding*. Orienta-konsultit, Helsinki, 1987
40. Kuutti K. Activity theory as a potential framework for human–computer interaction research. In: Nardi BA (ed.). *Context and consciousness: activity theory and human computer interaction*. MIT Press, Cambridge, MA, 1996
41. Keil M, Carmel E. Customer–developer links in software development. *Commun ACM* 1995;38(5):33–44
42. Nuseibeh B, Easterbrook S. Requirements engineering: a roadmap. In: *Proceedings of the international conference on software engineering*, 2000
43. Goguen JA, Linde C. Techniques for requirements elicitation. In: *Proceedings of the IEEE international symposium on requirements engineering*, 1993, pp 152–164
44. Carmel E, Whitaker RD, George JF. PD and joint application design: a transatlantic comparison. *Commun ACM* 1993;36(4):40–48
45. Bekker M, Long J. User involvement in the design of human–computer interactions: some similarities and differences between design approaches. In: *Proceedings of the HCI'00 conference on people and computers XIV*, 2000, pp 135–147
46. Kensing F, Simonsen J, Bødker K. MUST: a method for participatory design. *Hum–Comput Interact* 1998;13(2):167–198
47. Wood J, Silver D. *Joint application design: how to design quality systems in 40% less time*. Wiley, New York, 1989
48. Flynn DJ, Jazi MD. Constructing user requirements: a social process for a social context. *Inform Syst J* 1998;8(1):53–83
49. Checkland P. *Soft systems methodology: rational analysis for a problematic world*. Wiley, New York, 1989
50. Kensing F, Simonsen J, Bødker K. Participatory design at a radio station. *CSCW: J Collab Computing* 1998;7(3/4):1–32
51. Checkland P, Scholes J. *Soft systems methodology in action: a 30-year retrospective*. Wiley, Chichester, 1999
52. Davidson EJ. Joint application design (JAD) in practice. *J Syst Software* 1999;45(3):215–223
53. Purvis R, Sambamurthy V. An examination of designer and user perceptions of JAD and the traditional IS design methodology. *Inform Manage* 1997;32(3):123–135
54. Introna LD, Whitley EA. Against method-ism: exploring the limits of method. *Inform Tech People* 1997;19(1):31–45
55. Nandhakumar J, Avison DE. The fiction of methodological development: a field study of information systems development. *Inform Tech People* 1999;12(2):176–191
56. Fitzgerald B. An investigation of the use of system development methodologies in practice. In: *Fourth European conference on information systems*, 1996, pp 143–161
57. Bellotti V. Implications of current design practice for the use of HCI techniques. In: *Proceedings of the HCI conference on people and computers IV*, 1988, pp 13–34