

INTERNATIONAL JOURNAL OF SOCIOTECHNOLOGY AND KNOWLEDGE DEVELOPMENT

January-March 2009, Vol. 1, No. 1

Table of Contents

RESEARCH ARTICLES

- 1 A Meta-Analysis of the Direction and State of Sociotechnical Research in a Range of Disciplines: For Practitioners and Academics**
Elayne Coakes, University of Westminster, UK
Jim Coakes, University of Westminster, UK
- 53 My Top 10 Lessons on Lessons Learned Systems**
Jay Liebowitz, Johns Hopkins University, USA
- 58 Mind the Gap**
David Tuffley, Griffith University, Australia

BOOK REVIEW

- 70 Socio-Technical Knowledge Management: Studies and Initiatives**
Reviewed by Dianne Willis

INTERVIEW

- 75 A Sociotechnical Conversation:**
Barbara Farbey and Frank Land

Mind the Gap

David Tuffley, Griffith University, Australia

ABSTRACT

The application of Socio-Technical theory seeks to improve the alignment between the technical and social sub-systems that comprise organisations. The developers who create the technical systems and the people who use the systems are manifestations of the socio-technical dynamic. Yet a gap exists between these two groups that create a sometimes strong dynamic tension that is a worthy subject for research. Despite many years of study, the User-Developer gap seems as pervasive as ever. This article explores the nature of the gap, and proposes some practical solutions.

Keywords: Socio-Technical theory; User-Developer gap

INTRODUCTION

The production of high quality software calls for close cooperation and understanding between users and developers, particularly during the requirements gathering and analysis stage. From a Socio-Technical point of view, the challenge is to recognize the interdependency of the stakeholders and to work towards creating the conditions in which the sub-systems work harmoniously with each other. Ways must therefore be found to close the User-Developer gap that disrupts interdependency. The

first step would be to better understand the nature of the gap.

Software developers tend to have a technological mindset, being inherently inclined that way from an early age. Years of technical education and on-the-job training then follow to develop their technical skills to a higher standard. The pursuit of technical excellence, for many of them, is a matter of professional pride. Conversely (the majority of) software users have a non-technical or limited technical view of the world. Their interaction with software is a means to an end, not an end in itself.

This dichotomy is the basis of the gap under discussion here.

Bringing users and developers into closer cooperation through improved mutual understanding is in the best traditions of Socio-Technical Design. There should be substantial user involvement in the system design process Scaacchi (2004). In the Socio-Technical context, systems are defined broadly. A system can be comprised of networks of users, developers, information technology at hand, and the environments in which the system will be used and supported (Scaacchi, 2004).

Achieving this closer cooperation and mutual understanding requires an understanding of the dynamics of the organisational culture in which systems development is performed.

HISTORICAL VIEW OF USER-DEVELOPER

Cultural Differences

Organisational culture is a useful context and perspective within which to explore the nature of the User-Developer gap. Culture is an organization's way of thinking about the world and itself, how to get things done, how to solve problems. Software developers living in a world of technology have their own cultures, their own deeply ingrained ways of doing things. Such a culture might find it difficult to readily understand, much less embrace the seemingly foreign culture of the

business user, with their unfamiliar priorities, preoccupations, and ways of doing things. It is a case of 'same planet, different worlds'.

An organisation develops its own unique culture over time, evolving through the stresses and strains of its day-to-day operations. An awareness of the mechanisms of organisational culture may well be limited. People simply go about their jobs in the way they have become accustomed to, and do not think much about it until an external threat to their security and continued existence is perceived. Having evolved more or less effective ways of protecting itself and getting on with business, a problem arises when IS project management is encouraged to use processes like Joint Application Development and Participative Design. Developers are often reluctant to do participative systems development because it is seen to be expensive and time-consuming (Feeny, Earl and Edwards, 1996).

It is perhaps for this reason that practices such as Joint Application Development (JAD), and Participative Design (PD) whose purpose is to facilitate user-developer cooperation are not practiced. While these practices are known to work when diligently applied, organizations often seem reluctant to change their ways. JAD and PD and other methods designed to bring about closer cooperation and understanding between users and developers are seen as contrary to the way things are done in a particular work setting, and are perhaps seen as being time consuming

and therefore expensive in a “time is money” commercial setting. This is the likely reaction in an organisation where the Developer as Systems Expert paradigm is dominant (Hirschheim and Klein, 1989). These are all good reasons to avoid using JAD or PD if you are a hard-pressed project manager.

Software developers apparently possess characteristics that inhibit their working relationship with other members of the organisation. As far back as 1991, Grindley surveyed IS directors and found that 46% reported that the culture gap between IS professionals and business counterparts was their most important challenge in terms of service delivery. 56% believed that the culture gap inhibits their organization’s ability to achieve strategic advantage using IS. 56% is a startlingly high figure. CIO’s or IS Directors are in a unique position to evaluate the effectiveness or otherwise of their department.

Grindley explains that the culture gap is manifested by users and developers having differing approaches to motivation, goals, language, and problem-solving. These differences brought about not only difficulties in communication, which is an overt manifestation of the gap, but also reveals that the mind-set is likely to be different. Mind-set can be said to be a covert manifestation of the cultural gap. Having different notions of goal-setting and problem-solving are indicative of these differences. Grindley’s findings are consistent with even earlier findings of researchers such as Edstrom (1977),

Gingras and McLean (1979), and Zmud and Cox (1979) when they reported on the distinctive ways of thinking and acting of IS professionals.

Taylor-Cummings and Feeny (1997) also highlight the existence of what they call the ‘cultural gap’ between IS developers and users, a factor which has been blamed for the failure of IT projects since IT projects first began in the 1950s. This cultural gap is widely acknowledged but poorly defined. Definitions of the culture gap are descriptions of the symptoms, rather than the culture gap itself. While it is not well defined, its existence is causing rising alarm among IS management. Taylor-Cummings and Feeny (1997) discuss a survey of IT directors in the UK, in which 47% of respondents said this culture gap is their biggest problem. 56% thought the gap was seriously hampering their organization’s efforts to gain strategic advantage.

Taylor-Cummings and Feeny (1997) attempt to define the culture gap between IS developers and users in terms of two metaphors - organizations as cultural systems, and organizations as political systems (Morgan, 1986). Taylor-Cummings’ and Feeny’s (1997) definition is based on concepts of cultures and sub-cultures, diverse interests, conflict and power. This is a useful definition in the sense that it frames the problem in terms of culture. Other definitions discussed in this section have focused on differences in the thought processes of participants, those cognitive elements that comprise differing mind sets. Tay-

lor-Cummings and Feeny recognize the explicit existence of organisational culture, with sub-cultural elements within the broader organisational culture who have competing priorities and interests that then lead to conflict as each sub-culture attempts to realize their own particular goals and priorities. Power is sought after as a necessary way of achieving goals.

Wang (1994, p1) defines the culture gap as

a conflict, pervasive yet unnatural, that has mis-aligned the objectives of executive managers and technologists and that impairs or prevents organizations from obtaining a cost-effective return from their investment in information technology.

This statement highlights the nature of the gap in terms of a misalignment of objectives. This is in agreement with Grindley's earlier study that discussed the problem in terms of different approaches to goal setting, problem solving and language. This misalignment of objectives causes impairment of an organization's achievement of cost-effective systems development because the two categories of stakeholder pulling in different directions.

In recent times this misalignment of strategic objectives, or the differences in language, approaches to goals and problem-solving, has been recognized and discussed in terms of the Business-Technology gap. Baster et al (2001) refer to Business-Technology gap (B-T gap) in

terms of technology specialists lacking the domain expertise to react rapidly to changes in the business environment, while business users lack the technology skills to maintain the systems. This is discussing a gap in which the two categories of stakeholder are unable to adequately recognize, understand and accommodate the needs and wishes of the other. Technology specialists are identified as having well-developed skills at implementing technology, but do not have sufficient understanding of the way in which the domain or industry operates at commercial level. Domain specialists on the other hand have well-developed understandings of the commercial realities of their industry, but have only a sketchy understanding of how technology can be applied strategically to believe organisational objectives.

Hornik et al, (2003) highlight that good communication between IS professionals, IS staff and IS users is critical to the successful completion of an IS development project. They point out that the ability to interact with all potential stakeholders in an organisation, to clearly document requirements, and to effectively express ideas has long been recognized by researchers and practitioners as critical success factors.

Mann (2002) notes that there is little literature on the gap between end-users and the IT department. It has been observed that IT personnel have different personality traits than does the general population, going some way towards explaining why there is a gap. Martin-

sons & Chong (1999), and Shore (1998) discusses the important skills required for working in an IT environment, particularly those needed for effective collaboration with end-users.

Categorizing the Gap

The gap has been categorized in no less than nine ways in the academic and practitioner literature (Mann, 2002 pp*):

1. **The Perspective Gap** when the point-of-view of one stakeholder group is incomplete or ill-conceived. Developers may lose sight of the necessity for Systems to provide value to the business by meeting evolving business goals and that the IT department is not the centre of the universe. Users sometimes lose sight of technology as being a tool, and not an end in itself.
2. **Ownership Gap**: where developers feels a sense of proprietary ownership over the infrastructure, while users feel ownership over the business processes, leading to the demarcation disputes and territorial conflict that strain the relationship and create misunderstandings and misconceptions. Users can get the impression that developers are technical elitists, and developers come to see users as reactionary detractors.
3. **Cultural Gap**: when the stakeholder groups display different traits, values, working behaviours, and/or priorities due to each group attracting certain kinds of person, or acculturates members in the group. Developers tend to be more introverted, analytical, using rational persuasion to influence others. Business users are usually more extroverted, intuitive and use more sophisticated influence strategies. Both users and developers tend to adopt the culture of their respective professions.
4. **Foresight Gap**: where one stakeholder group has greater insight into how the future might unfold, but is unable to communicate that vision convincingly to the other stakeholder group. Developers may be well placed to foresee that a user proposed solution cannot work from a technical point-of-view. Alternately users may be better at determining that a developer proposed solution will not be acceptable to them, or will have a negative impact on some aspect of their operations.
5. **Communication Gap**: where one stakeholder group simply fails to understand what the other is saying. It is often said by users that Developers have an impenetrable jargon, yet it is also observed that the users may well have their own well-developed jargon. Developers find it difficult to translate the user needs of business units into useful productive systems because they do not understand the business processes and underlying rationale for them..

6. **Expectation Gap:** where users have unrealistic expectations about what developers are feasibly able to do. Users have come to expect more from systems because they have generally become more computer literate, or because they have become accustomed to the sometimes heroic efforts of developers to deliver the goods. At the same time, developers are sometimes known to make overblown claims as to what they can deliver, expecting all users to be technologically naïve.
7. **Credibility Gap:** where the past performance of developers has been substandard. This is often attributable to failed development projects, or poor customer service such as a not very helpful helpdesk. From the developer's perspective, they may have found users to be overly demanding and/or resistant to change.
8. **Appreciation Gap:** where one stakeholder group implicitly feels unappreciated by the other. Developers may form a view that their hard work, long hours and contributions to the organization go unappreciated except when something goes wrong. There is some suggestion that developers, in some cases, wish to be more involved in business planning, but are not invited to do so.
9. **Relationship Gap:** where the stakeholder groups do not interact with sufficient frequency to be

able to form a viable, constructive relationship as the basis for ongoing work. This might be reinforced by entrenched preconceptions about the other group.

Communication is implicated to some extent in all nine categories above. The Communication Gap (No. 5) is explicitly about communication, while the Perspective Gap, the Ownership Gap and the Foresight Gap and others might be describing inner states of mind; they are externalised to other parties by the communication process.

Poor communication is cited as one manifestation of a cultural gap between stakeholders (differing approaches to motivation, goals, language, and problem-solving, Grindley, 1991). Taylor-Cummings' and Feeny (1997) discusses the nature of the gap in terms of cultures and sub-cultures, diverse interests, conflict and power. Communication is a consistent factor in each of these elements. Good communication enables the reconciliation of diversion and interests, poor communication makes the problem worse. Wang (1994) discusses the nature of the culture gap in terms of a misalignment of the stakeholder's objectives. The communication process necessarily mediates the formulation of objectives and strategic plans.

An examination of these categories highlights the importance of developing effective communications strategies, an area that will be examined the next section.

User-Developer Communication

The Information Systems literature is divided on whether user participation is desirable on development projects. Gallivan and Keil, (2003) typify one view, that users who are more involved during the requirements determination process are more likely to feel they have a stake in the system, and therefore are more likely to be satisfied with the system. On the other hand, Ives and Olsen (1984) in their oft-cited meta-analysis of 30 empirical studies where user participation was a key variable came to the conclusion that there was insufficient evidence that user participation provided benefits.

It is often implicitly assumed that any user participation is beneficial, yet this is not necessarily the case. Gallivan and Keil (2003) suggest that project managers and software developers must look beyond the information content that users provide. They should also investigate what information users may not be readily volunteering, and try to create an environment in which users feel free to openly discuss their concerns, regardless of whether these concerns are positive or negative in character.

Gallivan and Keil (2003) acknowledge that user participation in system development is critical to the successful design and implementation of systems, but that it is dangerous to assume that user participation necessarily leads to successful project outcomes. They discuss the concept of power asymmetry

between developers and users. Where developers had sanctionary power over users, users would be less inclined to adopt a system that they had jointly designed developers, even when the users had explicitly agreed to use the said system. Ineffective communication occurs when developers have this sanctionary power over users. It makes it more likely that they (the user) will not use a conjointly designed system (ibid).

User-developer communication strategies can be seen as a function of organisational culture. It might therefore be useful to examine an appropriate model for understanding organisational culture such that effective communication strategies might be developed.

The Gap in Practice

An illustration of the gap in practice is seen in the following example. The developer group with their technical mindset is represented by the small organisation. They are embedded in the larger organisation as an expert technical group performing software development and upgrades on the larger organisation's information systems. Their relationship had been ongoing for at least three years at the time of the study. The data derives from a 12 month action research project performed by the author.

In this example, Johnson and Scholes (1999) model of organisational culture is used to illustrate the gap. Specifically, their six elements of or-

ganisational culture form the basis for the analysis of the gap. The elements are described as follows:

1. **Formal organisational structure**, or the more informal ways in which the organisation works, reflect power structures and delineate important relationships
2. **Stories**. The stories related by individuals that embed historical events in the present and highlight important events and personalities.
3. **Symbols** like logos, offices, cars and titles; or the type of language and terminology commonly used; and which become a short-hand representation of the nature of the organisation.
4. **Routines and Rituals**. The routine ways that member of the organisation behaves towards each other and that link different parts of the organisation. Rituals include training programmes, promotion and judgments that indicate relative importance in the organisation.
5. **Control systems**, measurement and reward systems that emphasize what is important in the organisation, and which focus attention and activity.
6. **Power structures**, the most powerful managerial groupings in the organisation are likely to be ones most associated with core assumptions and beliefs about what is important.

The following table illustrates how the cultures of developer group (small

organisation) differed from that of the larger, client organisation:

It can be seen from this table that on all six dimensions a continuum exists in which the large and small organization's occupied somewhat opposing ends. Hierarchical versus flat management structures, formal versus informal routines and rituals etc. Communications strategies that take account of this variability need to be developed to reconcile these differences if in an effective working relationship is to be achieved.

The results of this analysis can be generalized to include organizational units within a larger organisation, such as the IT development group and their relationship with other business units.

Facilitator of Communication to Bridge User-Developer Gap

Table 1, above, highlights the differences in the organisational cultures of developers and users, which go towards explaining some of the difficulties experienced by the stakeholders of IT development projects. Users with their general, non-technical perspective and developers with their almost exclusively technical perspective do not inhabit the same worlds as each other, despite their being in the same physical location. They do not share a common world view and consequently do not possess a common language with which to describe their worlds to each other. A facilitator of communication to act as a bridge, an interpreter between the

Table 1. Summary of cultural differences using Johnson & Scholes 1999 model

Element	Large organisation	Small organisation
Structure	hierarchical	flat
Stories	past/present	present/future
Symbols	affluent	economical
Routines/ Rituals	formal	informal
Control	punishment	reward
Power Structures	paternalistic	entrepreneurial

stakeholder groups is needed to bring about this common understanding and effectively close the gap.

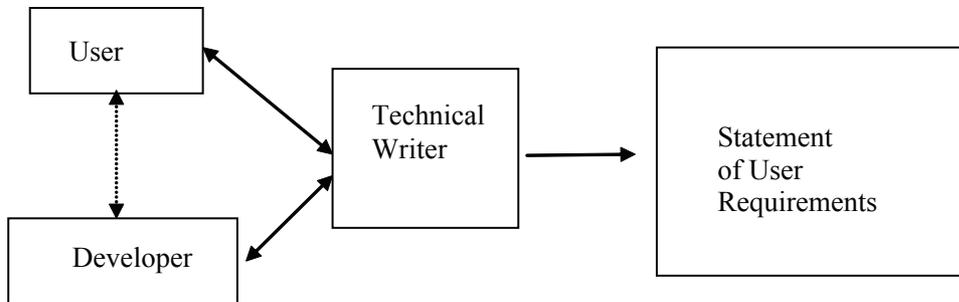
The nature of the gap has been explored and categorized in this article, and is seen in the context of organisational culture. Johnson and Scholes' (1999) model provides a conceptual basis upon which to think about organisational culture, and to perform an analysis that might be helpful when developing communication strategies to bridge the gap.

Using Johnson and Scholes' model to analyse the differing natures of two organizations interacting with each other over a 12 month period during a software development project shows considerable difference between the two cultures, highlighting the need to place an effective bridge across the gap in the form of a facilitator of communication. The study focused on an independent contracting organisation performing development services within the larger organisation. To some extent this has polarized the results in the way seen in table 1 above.

The evidence of the research project that gave rise to the above summarized results points clearly to be benefits of having an existing member of the software development team act as a facilitator of communication between developers and users. These benefits include deriving a clearer and more comprehensive understanding of the user requirements that can then form the basis of the developed system. This avoids the costly problem of discovering shortcomings in functionality after the system has been implemented.

Where available, business analysts traditionally perform the role of facilitator of communication between the business interest groups and developers. But business analysts are not present on every project. In the absence of a business analyst a technical writer might effectively be substituted in the role. The technical writer's ability to understand both technical and non-technical perspectives makes them well-qualified for this task. The technical writer also poses little threat to the organisational culture of the software development

Figure 1. Technical writer as facilitator of communication



team, and is therefore more likely to be accepted, even welcomed as someone willing to take on the task of developing the requirements specification document. A deceptively simple solution to a complex problem

Social-Technical Gap

The User-Developer gap is a manifestation of the broader Social-Technical gap. It is a deficit between what the users want and what the technologists have produced. From the evidence presented in this article, it is reasonable to conclude that system success is contingent on finding ways to bridge this gap that has been characterised in nine ways, all of them concerned with aspects of perception and belief among the stakeholders that result in misunderstanding and perhaps conflict. This is evidently a social process. It may therefore be asserted that a successful software system is a *social* system that has been technically implemented, and less a *technical* system that has been socially implemented.

A statement like this is at first glance a controversial one. Is not a computer system a technical system?

Thus this article argues that a *software system is a social system that has been technically implemented*. In support of this statement, we need only observe that in reality what comprises a system is the collection of users, interacting with the software/hardware and each other to produce the outputs required by the sponsoring organisation. Without these socially-constructed agents known as users producing outputs that serve the organisation's needs, the system is little more than technical infrastructure waiting for user input. The technical infrastructure exists to serve the needs of the user; it does not, in the majority of cases, exist for its own sake.

For as long as developers continue to regard computer systems as primarily technical when it is actually a social system technically implemented, the social-technical gap is likely to continue. It may be unrealistic to expect developers

to change the way they think, at least in the short-term, given the strength and durability of entrenched organisational culture. Nor is it realistic to expect users to become more technically literate, able to speak the technical language of the developer. What is needed is a facilitator of communication, a kind of interpreter who can mediate between users and developers. In practical terms this could be a business analyst, technical writer or project champion. Whoever they are, they are essential if the User-Developer gap is to be bridged.

REFERENCES

- Baster , G., Konana , P., Scott, J., (2001) *Business Components: A Case Study of Bankers Trust Australia Limited*, Communications of the ACM, 44(5), pp.92-98.
- Edstrom, A. (1977). *User influence and the success of IS projects: a contingency approach*. Human Relations, 30(7), pp.589-607.
- Gallivan, M.J., Keil, M., (2003). *The User-to-Developer Communication Process: A Critical Case Study*. Information Systems Journal 13, pp.37-68.
- Gingras, L., & McLean, E.R. (1979). *A study of users and designers of information systems* (IS working paper 2-79). California: Graduate School of Management, UCLA.
- Grindley, K. (1991). *Managing IT at board level: The hidden agenda exposed*. London, Pitman.
- Hirschheim, R., & Klein, H.K. (1989). *Four Paradigms of Information Systems Development*, Communications of the ACM, 32(10), pp.1199-1216.
- Hornik, S., Chen, H-G, Klein, G., Jiang, J., (2003). *Communication Skills of IS Providers: An Expectation Gap Analysis From Three Stakeholder Perspectives*. IEEE Transactions on Professional Communication, Vol 46, No. 1. pp.17-29.
- Ives B., Olson M.H. (1984). *User Involvement and MIS Success: A Review of Research*, Management Science, Vol. 30, No. 5, May, 1984, pp.586 - 603.
- Johnson, G., & Scholes, K. (1999). *Exploring Corporate Strategy*, 5th edition. London, Prentice Hall.
- Mann, J., (2002). *IT Education's Failure to Deliver Successful Information Systems: Now is the Time to Address the IT-User Gap*. Journal of Information Technology Education, Vol 1, No. 4, pp.255-256.
- Martinsons, M. G. & Chong, P.K.C. (1999). *The influence of human factors and specialist involvement on information systems success*. Human Relations, 52, pp.123-152.
- Morgan, G. (1986). *Images of organisation*. London: Sage Publications.
- Scaacchi, W., (2004) *The Encyclopedia of Human-Computer Interaction*, W. S. Bainbridge (ed.) Berkshire Publishing Group, 2004
- Shore, B. (1998). *Managing End-User Challenges*. Information Systems Management, 15, 79-83.
- Taylor-Cummings, A., & Feeny, D.F., (1997). *Managing IT as a strategic resource*, London: McGraw Hill, 171-198.

- Wang, C.B. (1994). *Techno Vision*, New York, McGraw-Hill.
- Zmud, R.W. & Cox, J.F. (1979). *The implementation process: a change approach*. MIS Quarterly, 3(2), 35-43.

David Tuffley is a lecturer in the School of ICT at Griffith University, and a senior consultant in the Software Quality Institute (the SQI is an Australian transition partner of CMU's Software Engineering Institute). Before academia, Tuffley consulted in the IT industry for 17 years beginning in London in the 1980's where he began as a technical writer and progressed from there to business analysis and software process improvement work in the public and private sectors. Tuffley has an undergraduate background in psychology and anthropology, and a postgraduate focus on software engineering, information systems and the socio-cultural aspects of computing.