Chapter 3
Leadership Can Bridge the User–Developer 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, practical solutions to the User-Developer gap still seem elusive. This chapter explores the nature of the gap, and proposes a leadership model that improves the capabilities of project managers and team members to bridge the 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
Leadership Can Bridge the User-Developer Gap

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 co-operation 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.

REVIEW OF USER-DEVELOPER CULTURAL DIFFERENCES

Chen et al. (2009) observe that users and developers have differing world-views that leads to difficulties in arriving at an accurate and comprehensive set of user requirements. They suggest that effective coordination between these stakeholders can mitigate the risks of developing software based on incomplete requirements (Chen et al, 2009). Effective models of co-ordination might therefore include a leadership model that enhances the management capability of project managers by adding the persuasive element of leadership to the mix (Tuffley, 2009, 2010). Such a model might also be used to good effect by team members to develop closer cooperation and higher performance. This paper outlines such a model.

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. And 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