The Role of the Software Life Cycle in the Development of Interorganisational Systems

Ana Maria Ramanath
Brunel University, ana.maria.ramanath@brunel.ac.uk

Ray J. Paul
Brunel University, ray.paul@brunel.ac.uk

Robert Macredie
Brunel University, robert.macredie@brunel.ac.uk

Follow this and additional works at: http://aisel.aisnet.org/amcis1997

Recommended Citation
http://aisel.aisnet.org/amcis1997/297
The Role of the Software Life Cycle in the Development of Interorganisational Systems

Ana Maria Ramanath, Ray J. Paul and Robert Macredie
email: {ana.maria.ramanath; ray.paul; robert.macredie}@brunel.ac.uk
Department of Computer Science and Information Systems at St John's
Brunel University
West London, Uxbridge
Middlesex UB8 3PH
UK

Introduction

Modern business organisations are increasingly concerned with developing, and subsequently maintaining, competitive advantage. To support this, business organisations are seeking to establish strategic partnerships with their customers, suppliers, and distributors. Implicit in such partnerships is the need to effectively employ information technology to support the timely exchange of salient information across the different organisational boundaries of the partners. Systems which support this type of information exchange are referred to as Interorganisational Systems (IOS). This is reflected in a common definition of IOS which notes that they are "information systems which allow companies to share data and software with trading partners across organisational boundaries" (Holland et al., 1994).

The importance of IOS is likely to increase as more and more organisations realise and exploit the potential of current information technology to support their dealings with customers, suppliers and distributors. Future developments in information technology - such as improvements in networking - are also likely to offer further potential for competitive advantage through IOS.

It would seem important, then, that the process of IOS development is well understood so that effective systems can be achieved. Yet, this does not seem to be the case, with only limited discussion of IOS development issues being represented in the literature. Instead, most IOS research tends to concentrate on the effects of strategic, organisational and environmental issues associated with IOS implementation in organisations.

IOS development issues that have been reported include proposed guidance (Konsynski, 1992) as to what to take into account when designing IOS for businesses. This research suggests that concentration on purely transaction interchange types such as EDI could often lead to little more than 'speeding up the mess'. Other research has considered how IOS could be used as an opportunity to rationalise business processes, not just in one organisation but across many organisations (Clark and Stoddard, 1996).

The diversity of the little research that is reported needs to be considered closely and developed to provide a better understanding of the issues and needs in IOS development. It is important that IOS development is considered in terms of its unique nature, considering the challenges of development across potentially diverse organisational boundaries. The appropriateness of traditional development models and methods of IS should be questioned and, at the very least, developed in an attempt to encompass issues raised by the context of IOS. It is against the background of the limitations of traditional IS methods for IOS development that this research is framed.

Research Framework

Because of the apparent lack of research which specifically addresses IOS development, and the subsequent scarcity of paradigms for IOS development in the literature, it could be argued that the obvious route for businesses to follow is that of using traditional Information Systems (IS) development methods. But the IS literature shows that system development methodologies, and the information systems produced by them,
have been the subject of much criticism. (Fitzgerald, 1996; Beath and Orlikowski, 1994; Avgerou and Cornford, 1993).

However, there is also an extensive body of research dedicated to finding new directions for IS development approaches. This has been exemplified by the studies of Boehm (1988), Mumford (1983), and Avison and Wood-Harper (1990), to name but a few. Something else that becomes apparent on examining the literature is that methodologies have almost as many critics as supporters. This point is well illustrated in the comprehensive industry survey about the use of methodologies carried out by Fitzgerald (1995).

In both established and emerging approaches to IS development, it could be argued that methodologies tend to succeed in providing a sort of high-level 'plan' for the guidance of the project. This is achieved through the division of the development process into stages, giving the impression that development is somehow more 'manageable'. Even if the resulting system fails to deliver what it is supposed to upon completion, this approach at least gives the stakeholders involved a feeling of being in control by following a plan.

Most methodologies reflect the steps of the Software Life Cycle (SLC), with very little differences between them (Davis, 1974). But, it could be argued that the SLC as a framework for the development of modern systems is deemed to be inadequate. The SLC model dates back to the necessarily reductionist approach of scientists involved in World War II computing projects. The problems addressed by such projects were highly deterministic. The solutions were also strongly linked to the (limited) hardware capabilities of the time (Agresti, 1986). In contrast, modern IS are fundamentally 'soft' human activity systems (Checkland and Scholes, 1990) and yet we are trying to develop them by using a 'hard' product-oriented approach embedded in methodologies. This dichotomy might constitute a major cause of systems development failure.

In this research, we are concerned with the appropriateness of the SLC as a framework for IOS development. This frames the research question: Why is the SLC needed for the development of IOS? IOS are by nature more multi-faceted (Lyytinen, 1989), and multi-functional than IS (Konsynski, 1992; Hopper, 1990; Meier and Sprague, 1990). This suggests that the appropriateness of the SLC model as a framework for the successful implementation and support of IOS is highly questionable. This research intends to assess the impact that the SLC has on the overall development, implementation and support of IOS. Its longer term aim is to suggest ways of alleviating the shortcomings of the SLC as a framework for IOS development.

**Methodology**

The main source of data collection in this research is the multiple case study strategy, as explained by Yin (1989). The first case study (which is on-going) is an in-depth study of a two year development project in a UK Sales organisation. The delivered system from this project was considered successful by the business. The study of the project will be carried out in parallel with the observation of the system in its current maintenance phase. A second case study, within the same organisation, will then further explore the research question, drawing on the experiences and findings of the first case study. The second case study aims to be a reconstruction of an IOS project whose outcome was not considered as successful as the IOS subject of case study one.

A third, and final, case study, at a separate UK organisation, is comprised of an IOS developed by an IT Vendor. The purposes of the system in this case was to provide technical support services to a large number of independent customers.

The main sources of data collection during the case studies will consist of semi-structured interviews with a sample of stakeholders from the participant organisations. Other sources of data collection will be direct observation, informal conversations, and associated system documentation and functionality.
Preliminary Observations

Most IOS design and adoption issues found in the literature appear to be in line with our experience as practitioners in IOS development. However, our research goes a step further to investigate issues regarding not only the design, but also the specification, development, implementation and support of IOS in businesses.

During our field experience developing IOS, we have observed certain peculiarities with regards to the development process. For example, the methods and tools to be used in the development of IOS requires the consensus of larger numbers of IS developers. However, considering that different IS functions in different organisations are biased towards formal or informal approaches for systems development, reaching consensus may not be an easy task. In addition to these, the expectations of the stakeholders involved need to be carefully managed throughout.

The user community itself tends to be large and heterogeneous. This means that representatives of different parts of the organisations involved are selected to be an active part of the development process. These representatives must agree on a minimum set of system functions, or else the resulting system could be very large and unmanageable. This usually leads to their information needs being covered only partially by the implemented IOS with all of the predictable consequences. On the other hand, a variety of reasons could lead users to leave the project at any stage, and this has an impact on the design and delivery timescales of the project.

Also, the project manager(s) plays a major role in the harmonious communication between the different stakeholders throughout the project life cycle. Hence, the task of managing a project of the magnitude of an IOS requires talented and skilled stakeholders and project managers. Finally, the geographic space of an IOS tends to be much wider than that of an IS, and considerations such as time zones, cultural backgrounds, long distance travelling, relationship with hardware and software suppliers need to be handled tactfully and efficiently throughout the IOS life cycle.

Implications and Contributions

This investigation reviews three IOS development projects in two business organisations in an attempt to assess the impact of the SLC on IOS development. The study has the potential to make important and interesting contributions relating to the overall process of specification, design, development, implementation and maintenance of IOS. As described above, there are shortfalls in current IS development methods which might make them unsuitable for IOS development. It is hoped that the results of this investigation will uncover some of the key issues to be considered by managers and developers when embarking on IOS projects and provide insights to inform the design of IOS.

Longer term aims of the research are to develop guidance for IOS developers and to explore the extent to which such guidance can be usefully formalised into prescriptive design methods. The usefulness of any such methods will depend critically on their flexibility in considering and addressing the specific nature of the organisations involved and the relationships between them which have to be 'captured' in the resulting IOS.

References


