Studying the Translations of NHSnet

EDGAR A. WHITLEY, London School of Economics and Political Science, UK
ATHANASIA POULOUDI, Brunel University, UK

This paper explores the ways in which innovative information systems projects take on a life of their own. The paper begins by reviewing some of the more traditional ways of making sense of this phenomenon: resistance to change, escalation and unintended results, before introducing the sociology of translation. This provides a theoretical framework for viewing the transformations that an information systems project undergoes. The framework is then applied to the case of the NHSnet project in the United Kingdom. Using the language of sociology of translation, we consider the underlying stakeholder relations in the case study and draw more general conclusions for the responsibilities of stakeholders involved in an information systems lifecycle.

INTRODUCTION

Few information systems projects follow a straightforward path from initial idea through to widely used working system. Instead, what typically occurs is that the nature of the innovation and the purpose of the project changes many times during the implementation process. Much information systems research attempts to try to explain what goes on over the life of the project. The purpose of this paper is to add one new element to the range of conceptual tools available to the information systems researcher trying to understand what happens to a particular innovation and to demonstrate how the insights from using this tool can add to our understanding of information systems implementation.

The paper begins by reviewing some of the main ways in which the changes that an information systems project undergoes have been conceptualised. These include unintended effects; resistance to change and escalation. The paper then introduces the notion of translation that has been used in the field of science studies and shows how it can be applied to the study of information systems, paying particular attention to the particular kinds of translations that an information systems project can undergo. The paper then presents the case study, namely the introduction of a new shared network in the UK national health service (NHSnet). This project is seen as a series of translations and the paper explores some of the main translations and discusses their implications for relevant stakeholders. The paper ends with a summary and discussion of the benefits of using this approach to analyse the “life” of information systems projects.

UNDERSTANDING THE LIFE OF A PROJECT

There are many different ways in which information systems researchers have tried to conceptualise the life of a project. One approach is to describe the events associated with a project and to talk about them in terms of anticipated, unanticipated and emergent changes. Another approach is to talk about the changes in terms of resistance to change and the mechanisms that can be used to counter the implementation of the system. A third approach is to consider the project as potentially escalating out of control.

Unanticipated Changes

Orlikowski (1996) describes an organisation introducing Lotus Notes as a groupware solution for a firm in the software industry. The firm, pseudonymously known as Zeta Corporation, is the developer of a range of powerful software products in the area of decision support and executive information. Their tools are based around the Omni fourth generation language and allow users considerable flexibility in how to analyse their data. As a consequence, many users have technical queries about how to make the products perform particular tasks. The groupware system was introduced into the product support area to enable the sharing of information about problems between the support team (Orlikowski, 1996, pp. 25–27).

The organisation had previously used a stand-alone system to store details about client problems. The existing system had limitations in terms of inconsistent usage, poor data quality and limited search capabilities. The intention behind the new system was to pool all the data in one, shared system. Thus advisors would be able to draw on the experi-
ences of all previous interactions, rather than just their own. As an illustration of the success of this, the number of records of client problems in the database grew from 4,000 records to 35,000 in the two years from December 1992. As Orlikowski notes, however, the system was successful, in part, because of the particularly cooperative culture in the department. Thus, if the same technology had been introduced into an organisation with a less cooperative culture, it is unlikely that a similar success would have been noted.

In describing the changes that arose as a result of the system, Orlikowski differentiates between anticipated changes, opportunistic changes (which are not anticipated ahead of time but are introduced purposefully as a result of an unexpected opportunity or event) and emergent changes which arise spontaneously out of local innovation. An example of an anticipated change arising from the system was the ability of managers to control the resources in the department more easily; by being able to monitor the number of calls they were able dynamically to change the allocation of work. An opportunistic change that arose from this was the decision to introduce the role of support partners who had specialist knowledge and who could support less experienced staff who handled the front line of calls. An unanticipated consequence was the way in which these front-line staff dealt with their new support partners. The organisation discovered that many junior specialists were reluctant to reassign calls to their support partners; often they felt that tackling difficult problems would help them to develop their own careers whereas on other occasions, the reluctance arose from a concern not to be seen to be dumping problems on their support partners.

Unfortunately, Orlikowski’s analysis goes no further than differentiating between the three types of change. No explanation is given for why emergent changes arise, how they could be prepared for and how they can be controlled.

Resistance to Change

A second way of conceptualising the changes that a project undergoes is through the notion of resistance to change. This is perhaps best typified by the classic paper by Keen (1981) which outlines a variety of approaches which have been used to counter the implementation of a new information system. Amongst the counter-implementation games identified by Keen are easy money, budget and territory whereby a project is supported because it can be used to support some needed activity within the sphere of influence (p. 29). Another game is tenacity whereby a project is kept incomplete until one’s particular terms are satisfied. Odd man out is used by players who give only partial support and withdraw when the project faces trouble (p. 29). Other games identified by Keen include up for grabs where a project with only lukewarm support is taken up by another player; reputation whereby a manager gets credit for being a bold innovator but leaves the project before the implementation stage and hence avoids any backlash arising from any problems that exist (pp. 29-30).

Thus, according to Keen, a project is under constant threat of counter-implementation and management must be prepared to take counter-counter implementation measures to ensure that the project succeeds. A similar argument is put forward by Markus (1983) who highlights the political aspects of any system implementation, seen from a perspective which emphasizes the effects of the interaction between the people and the systems.

Escalation

A third approach to understanding the phenomenon is through the notion of escalation. Keil (1995) defines escalation as a continued commitment in the face of negative information about prior resource allocations coupled with uncertainty surrounding the likelihood of goal attainment.

In order to study the factors that can lead to escalation, Keil describes the experience that ComputSys (a pseudonym) had with a project called Config. Config was a rule based expert system that was designed to help the company’s sales force produce error-free configurations prior to producing pricing estimates. Previously the company had made estimates based on incorrect configurations and had to bear the cost of any discrepancies itself. The organisation had had positive experience of another system (Verifier) which was used to produce correct system configurations and was therefore expecting that this project would be successful as well. The Config project was finally terminated 13 years after it was initiated. During this time, feedback about the project was predominantly negative. Eight years after the project was initiated, usage of the system had dropped to less than 2% of all transactions.

A number of explanations were given for the continued support of the project in the face of such negative assessments. Amongst the key arguments identified by Keil are the fact that the project was perceived to have a large net present value, that the project was regarded as an investment in research and development and that the problems appeared to be temporary setbacks rather than fundamental problems of concept.

Moreover, the organisation had a history of successful projects in this area, and the manager of the project was taking a high degree of responsibility for the success of the project. Indeed, Keil argues that the involvement of the strong project champion meant that the project was defended at times when it might legitimately have been cancelled.

Summary

Clearly there is overlap between each of these approaches; for example, what one sees as an unanticipated change could be viewed by another as an attempt at counter-implementation. This again could be seen by another as a project that is potentially escalating out of control. What all