Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1998 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1998

The Impact of Technology Characteristics on Infusion of Software Development Tools

Ravi Patnayakuni
University of Melbourne

Arun Rai
Georgia State University

The Impact of Technology Characteristics on Infusion of Software Development Tools

Ravi Patnayakuni
Department of Information Systems
The University of Melbourne

Arun Rai
Department of Decision Sciences
College of Business
Georgia State University

Abstract

Innovation diffusion research is a widely accepted theoretical basis for studying IT implementation but has many limitations. In view of these limitations some researchers have proposed that infusion of IT innovations in organizations should be studied. This paper empirically studies the impact of technology characteristics on infusion of systems development tools in organizations. The results indicate that while subsystem compatibility and scope of development tools strongly influenced infusion, methodology compatibility had no impact on infusion

Introduction

Information systems (IS) are integral to the management of day-to-day operations in most organizations. These systems are increasingly complex and often need to be developed rapidly in a short period of time. Software development tools may support one or more tasks in systems delivery and are expected to improve the quality of systems delivered as well as systems development productivity. At the same time evidence on performance gains has not been forthcoming It has been argued that one of the reasons for the performance shortfall is that ISDs do not make appropriate changes in their systems delivery process after acquiring such tools). Changes in workflow and in the systems delivery should accompany the implementation of new software development technologies.

Innovation diffusion research is a widely accepted theoretical basis for studying IT implementation but has many limitations. In view of these limitations some researchers have proposed that infusion of IT innovations in organizations should be studied (Cooper and Zmud 1990). This paper theoretically develops and operationalizes the measure of infusion in the context of systems delivery. It then identifies critical technology characteristics based on innovation diffusion theory that are likely to influence infusion of software development tools in organizations. Hypothesis are developed, empirically validated by and conclusions drawn for managers and for future research.

Infusion and the Software Delivery Process

Infusion was measured by identifying the theoretical basis for redesign of work systems in organizations by drawing from BPR literature (Hammer 1990). A framework was developed for identifying the locus of redesign in an organizational process by taking inspiration from Swanson's (1994) tri-core model for an organization. The model proposed technical, administrative, and informational cores as three cores of an organization, and was adapted to characterize the dimensions of a work system. It was proposed that to complement the capabilities of an innovation, all three cores of the organization need to be redesigned, characterizing infusion of an innovation. Items developed for measurement of infusion were based on the principles of reengineering and an analysis of the systems delivery process.

The three cores of the tri-core model represent the dimensions along which infusion of IT innovations can be assessed. Using the principles of reengineering it is possible to define infusion of IT innovations in each of the three cores. Infusion in the technical core is defined as 'the degree to which tasks are to focus on the efficiency and effectiveness of processes'. Automation, removal of redundant tasks and processes and reengineered workflow would characterize infusion in the technical core. Infusion in the administrative core is defined as 'the degree to which collaboration and empowerment are present'. Use of teams, group reward structure and multi-skilled personnel would mark infusion in this core. Infusion in the informational core is defined 'as the degree to which management processes are fact-based'. Infusion in the informational layer of the organization would be indicated by the continuous observation of transactional environments and their integration into decision making.

Software Development Technology Characteristics and Infusion

A development tool portfolio is a complex multi-dimensional technological system that provides support for software development tasks performed by individuals and project teams throughout the lifetime of an application system. ISDs use a variety of development tools that support one or more phases of software development. The popularity of client / server computing architecture and the trend towards Graphical User Interface (GUI) based environments has only added to the variety of development tools available in the market. Accordingly, scope, compatibility with development methodology, compatibility across tool(s), and system restrictiveness is used to characterize the development tool portfolio. The research model proposed a positive relationship of all the variables except methodology compatibility with infusion.

Development tools differ in their functionality and extent of support offered for different activities in the software development process from front end to back end to reverse engineering tools. As the scope increases it is likely to influence a greater number of activities in the systems delivery process. This is likely to increase the scope of redesign to take advantage of technology capabilities.

H_1: There is a direct effect (positive) of scope of an ISD's software development tool portfolio on their infusion.

Portfolios of development tools may also vary in the nature and philosophy of support. Methodology compatibility is defined as the fit between the orientation and approach of the development tool portfolio and methodology used for developing systems. A more compatible methodology would mean that the organization has already in place a process for software delivery that is not very much different from the one supported by the portfolio of development tools. Under the circumstances it is quite likely that the organization will make no changes to its work system and essentially use the automational capabilities of the portfolio and overlook any other capabilities it may have. The automation of existing tasks will thus further entrench the current systems delivery process in the organization. This is contrary to what both infusion and BPR would imply. To exploit the functionality of the development tool portfolio, work systems need to be redesigned. An opportunity for this would be created by the incompatibility of the development methodology in use to that of the development tool portfolio.

H_2: There is a direct effect (negative) of methodology compatibility of an ISD's software development tool portfolio on their infusion.

A development tool portfolio usually consists of several tools that need to work together to support systems development. Positive relationship of compatibility between development tools and infusion highlights the importance of tool and object integration in a development tool portfolio. Integration of systems development environment may be of two types; tool and object (Mi and Scacchi 1992). Tool integration provides a development tool set for systems development activities. Object integration would provide a consistent view of development artifacts and easy-to-use interfaces for generation, access, and control of these artifacts. Subsystem compatibility to ensure tool and object integration is determined by the interface of development tools with other tools in the portfolio. Compatibility within a portfolio allows development of greater interrelatedness among software delivery processes. In the absence of compatibility it would be difficult to design a software delivery process that exploits the capabilities of each tool in the portfolio in an optimal fashion.

H_3: There is a direct effect (positive) of subsystem compatibility of an ISD's software development tool portfolio on their infusion.

System restrictiveness of a development portfolio represents limitations placed on the type and sequence of tasks and activities that an ISD can undertake in its systems delivery process (Silver 1991). It also includes restrictions placed on user inputs to the development process. Restrictive tools are likely to be inflexible and difficult to adapt to the context in which they are used. Such tools are likely to be received negatively by developers, as they cannot be adapted for use. Further, restrictiveness makes it difficult to make use of all the functionality of development tools thus limiting the possibilities of redesign and as a consequence infusion of the technology.

H_4: There is a direct effect (negative) of system restrictiveness of an ISD's software development tool portfolio on their infusion.

Data Collection

The second phase of the study consisted of data collection by a self-administered mail survey. Development and validation of the instrument was done along the lines suggested by Churchill (1979). A directory of senior IS executives compiled by a market research firm was used as the sampling frame for this study. A total of 678 manufacturing and service organizations received the survey, of which 123 responded with an effective response rate of 18%.

Hypotheses Testing

Each hypothesis was tested using simple regression (Table.1). It was found in the study sample that as the scope of development tool portfolio increased, greater was the extent of infusion. The difference was statistically significant between ISDs with a full-life cycle portfolio and other ISDs, but not significant between ISDs with a multiple-phase portfolio and those that had a single-phase portfolio.

A negative relationship between methodology compatibility and infusion was observed but hypothesis (H_2) was not supported. The analysis indicates that an integrated development tool portfolio indicating high subsystem compatibility provides strong support for new and redesigned work processes. Hypothesis (H_3) was supported. A four-item scale developed to measure restrictiveness did not demonstrate adequate measurement properties and hence, was not used for testing the hypothesis.

Table 1. Summary of Results

Variable	Infusion
Scope of development tools	Supported (H_1)
Methodology compatibility	Not supported (H_2)
Subsystem compatibility	Supported (H_3)
System restrictiveness**	Not tested (H_4)

p<=.01 level of significance

Conclusion

The results suggest that multiple phase tool portfolios may support the development process but tend to perpetuate independence between different phases by not using technology to integrate them. Overall, the difference between the two ends of the continuum does support the hypothesis that full-life cycle portfolios are more likely to be associated with infusion and consequently a systems delivery process more in tune with the capabilities of the portfolio. The absence of a statistically significant relationship between methodology compatibility and infusion shows not only a lack of support for the hypothesis but the analysis also demonstrates the absence of a positive relationship. In developing the hypothesis it was argued that the presence of compatible methodology might not promote changes in the systems delivery process. Potentially, methodology compatibility can perpetuate existing work systems without resulting in benefits from the new technology. Restrictiveness is a complex construct that could possibly have more than one dimension, for instance sequential restrictiveness, decision restrictiveness, or input restrictions. A more complex scale, with multiple dimensions may be needed to capture this concept.

References

Churchill, G. A. 1979. A Paradigm for Developing Better Measures of Marketing Constructs. *Journal of Marketing Research*, Vol. 16 : pp. 64 - 73.

Cooper, R. B. and Zmud, R. W. 1990. Information Technology Implementation Research: A Technological Diffusion Approach. *Management Science*, Vol. 36, pp. 123 - 139.

Hammer, M. 1990. Reengineering Work: Don't Automate, Obliterate. *Harvard Business Review*, Vol.68, No. 4, pp. 104-114.

Mi, P. and Scacchi, W. 1992. Process Integration in CASE Environments. *IEEE Software*, March, pp. 45-53.

Rai, A. and Howard, G. 1994. Propagating CASE Usage for Software Development: An Empirical Investigation of Key Organizational Correlates. *Omega*, Vol. 22, No. 2, pp: 133-147.

Silver, M. S. 1991. *Systems that Support Decision Makers: Description and Analysis,* New York: John Wiley and Sons.

Swanson, E. B. 1994. Information System Innovation Among Organizations. *Management Science,* Vol. 38, No. 9.