

# THE LINK BETWEEN BPR, EVOLUTIONARY DELIVERY AND EVOLUTIONARY DEVELOPMENT

**Leslie Johnson**

**Computing Laboratory, University of Kent,  
Canterbury, Kent, CT2 7NF**

and

**Maria Stergiou**

**Computing Laboratory, University of Kent,  
Canterbury, Kent, CT2 7NF**

## ABSTRACT

In this paper we intend to show how the challenges of managing a Business Process Reengineering (BPR) project are consistent with the ones of a Systems Development project. As traditional management techniques were no longer appropriate in the changing business environment, companies employed BPR to achieve elevated business performance. Similarly, as traditional systems development approaches delivered disappointing results, system developers experimented with other models, including Evolutionary Delivery and Evolutionary Development, in order to enable successful technology exploitation by businesses. Both these business and systems initiatives embrace elements of cultural change, management flexibility, empowerment, organisational readiness, and technology introduction in a changing environment. We will present the similarities of the two initiatives and show how progress in one initiative could contribute to the progress of the other.

**Keywords:** Business Process Reengineering, Evolutionary Delivery, Evolutionary Development, Meta-Models.

## INTRODUCTION

The world is changing greatly from day to day and new elements are added to an already long list of considerations any company should address when developing its strategic objectives [1]. Some of these elements are new to the current business environment (e.g., sophistication of customer, people demanding fulfilment and personal meaning at work, green issues, collapse of middle management) and some already existed but have recently intensified (e.g., the fierce competition, increasing importance of business ethics and the lack of these in some businesses, the move away from unitary organisations toward federal, franchising and networking models, the 'global village', being a socially responsible company).

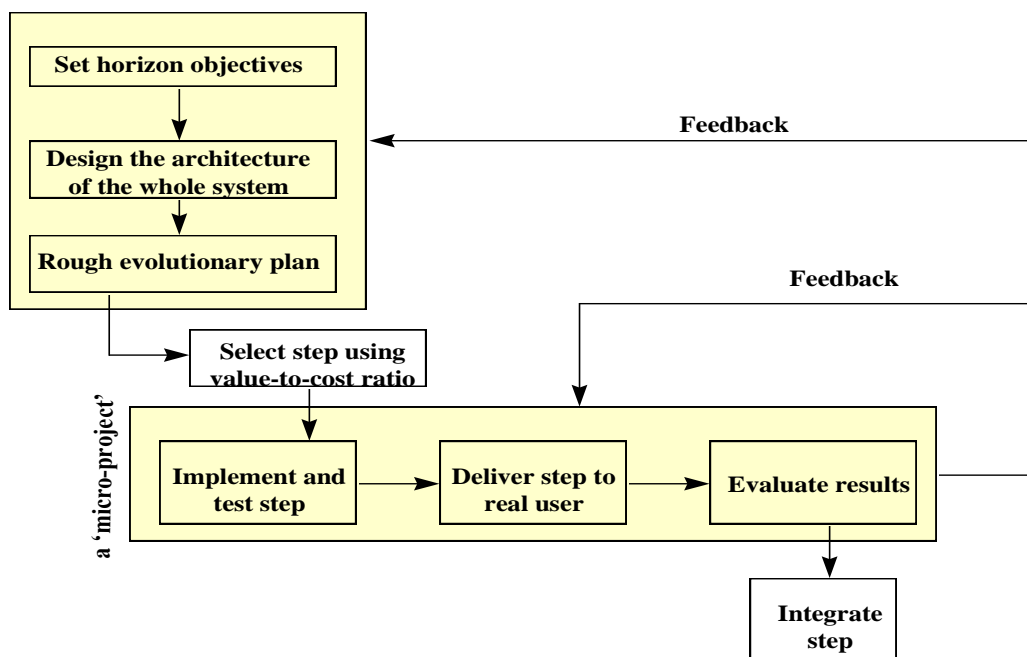
Throughout the eighties, the above mentioned elements in the business world, questioned traditional management behaviour and practices. This dynamic business environment called for refocusing on management thinking and as a result, management 'gurus' around the world came up with management tools and behaviours that would qualify a company to survive and successfully compete in the new

business era. Concepts like Total Quality Management (TQM), Just-In-Time (JIT), Downsizing, Business Process Reengineering (BPR), emerged; their purpose was formulated, and a “methodology” was quickly attached to them.

In such a business environment, heavy investment in Information Technology (IT) had delivered disappointing results. Hammer

methodologies delivered systems that failed to meet the needs of both senior management and end users alike since they were only involved when the final system was delivered [3].

These methodologies entailed that the systems were developed in an output-driven process. This meant that there was no conceptual room in the methodology to accommodate changing



**Figure 1: Evolutionary Delivery**

Adapted from T. Gilb, and S. Finzi, *Principles of Software Engineering Management*. Addison-Wesley, Wokingham, 1988

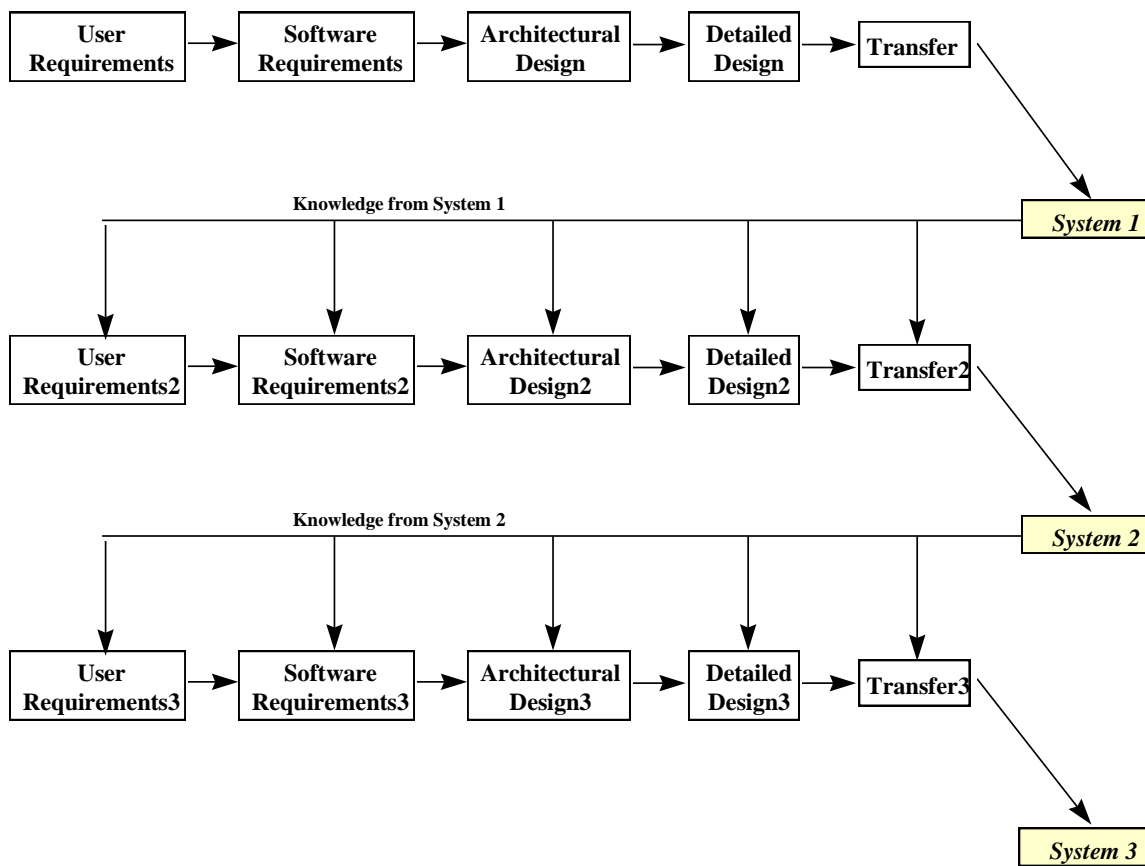
[2] gives two reasons for the disappointing results. The first reason is that companies tend to use technology to mechanise old, and possibly cumbersome, ways of doing business that have already proven inadequate. The second reason is that most IT applications were built applying traditional step-by-step system development methodologies. Such

requirements discovered in the development process. As a result, these changing requirements could neither be captured in the development process nor addressed by the monolithic system delivered. Such methodologies failed to acknowledge that business requirements continue to evolve

during the systems analysis, design [and maintenance] phases [4].  
 As traditional systems development approaches delivered disappointing results, system developers experimented with Evolutionary Development and Evolutionary

something to a real end-user on-site; measure the added value to the user in all critical dimensions; adjust both design and objectives based on the end-users' feedback.

The complete project is divided up into



**Figure 2: Evolutionary Development**

Adapted from R. Barillere and C. Esciihuela, OOR&D Day, CERN, Geneva, November 3, 1995

Delivery to enable successful technology exploitation by businesses.

**SYSTEMS ENGINEERING**

**Evolutionary Delivery**

Evolutionary Delivery is a software development methodology based on the following simple principle [5]: Deliver

potential steps. The steps with the highest ratio of user-value to development-cost are selected for early implementation (Figure 1). In other words, the steps are prioritised based on the minimum development effort that delivers the highest payoff to the end-users. When the feedback from the implemented step(s) is received then objectives, design,

user-value and cost are re-appraised and adjusted if necessary.

With Evolutionary Delivery the project evolves through steps which are continuously adjusted to meet the changing requirements of the end-users.

### **Evolutionary Development**

Evolutionary Development is a software development methodology. The objective of this methodology is to deliver a flexible and expandable core system. When requirements change during the system development process, a modified system that fulfils these requirements can be designed and developed with minimum time and effort (see Figure 2). The underlying principle is to design systems that are easily and quickly modified in the light of emerging requirements. The goal is to move from the design and implementation of static systems to the development of evolvable application families [6].

In Evolutionary Development, system “evolution” can take many forms, from accommodating a quick fix to a moderate or full upgrade to a complete customisation to particular business requirements. The aim is to develop each evolved system by investing minimum resources.

### **Evolutionary Delivery and Evolutionary Development are not Methodologies**

The origins of both Evolutionary Development and Evolutionary Delivery are in the information systems world. They are both presented as methodologies. However, they both are based on underlying principles rather than step-by-step approaches. Principles (i.e. guiding rules) outline what is to be achieved - not how. They are thus not best thought as methodologies - both are approaches to formulating systems development strategy.

## **BPR**

### **What is BPR?**

BPR is the fundamental rethink and radical redesign of business processes to achieve dramatic improvements in critical contemporary measures of performance, such as cost, quality, service, and speed [7].

### **BPR is not a Methodology**

Hammer’s famous statement, ‘Don’t automate, Obliterate’ (see [7] above) has one underlying message: there is no cookbook approach, no 10-steps-to-success plan, no manual for BPR. The people that cite, address, or deploy BPR as yet another management technique, have misunderstood the real value of it. BPR is not like TQM, JIT, Downsizing, or any other management tool of our century.

Methodologies like these have inherent faults, in-built shortcomings: they emphasise some elements of a business and provide the silver bullet for their best performance. TQM emphasised quality of product/service and ways to achieve it. JIT focused on minimising non-necessary activities in a customer-supplier relationship and ways to achieve it. BPR addresses all elements of a business but it does not specify ways to achieve their best performance. It only offers guidelines and some tools and techniques; the creative design and programme of change are provided by the practitioner.

In other words, because BPR is not a methodology, it appears that all you can do is start with a clean sheet of paper and follow the principles of reengineering. An organisation should have a vision of how it can be reengineered. BPR practitioners know the principles they have to follow and the tools and techniques that are available to them. Using these, and with the help of senior management, they initiate changes towards realising the organisation’s vision.

Thus, we claim that BPR is not best thought as a yet another methodology but rather as an

approach to business strategy; it offers an alternative perspective to formulating business strategy [8].

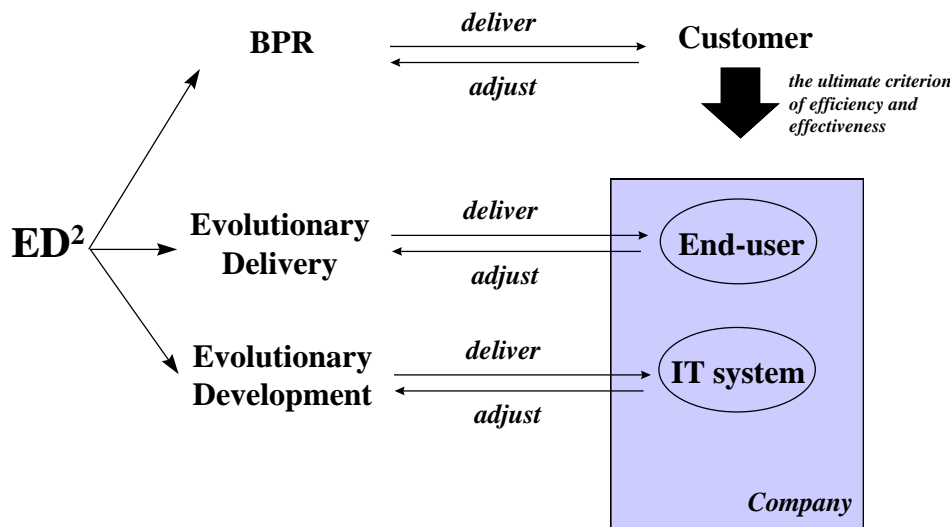
**SYSTEMS ENGINEERING MEETS BPR**

**The Common Challenges of BPR, Evolutionary Delivery and Evolutionary Development**

It is generally understood that IT must support strategy, people and processes. Further, IT is considered to be an important enabler of organisational transformation. Traditional IT has been seen as a service function within an organisation and not of strategic import. The

user computing and the advent of BPR, software development methodologies have remained rooted in the conceptual mental set of traditional computing.

It is clear that Evolutionary Delivery and Evolutionary Development, although motivated by software engineering concerns, are potentially more responsive to the organisation's interests. If their strategic apex is driven by strategic needs, their computational base can more easily be aligned to business strategy. BPR is the most mature candidate for aligning strategy, people and



**Figure 3: The ED<sup>2</sup> Model**

focus of IT was to largely mechanise clerical activities within functional departments rather than introducing company-wide system solutions. Established mainframe computing and the associated software development methodologies grew in that context and thus re-enforced an already rigid and limited deployment of IT. Despite the growth of end-

processes. Therefore we suggest that some form of BPR is the appropriate framework for the strategic apex of system development methodologies.

To integrate Evolutionary Delivery, Evolutionary Development and BPR, we

propose the construction of a meta-model of systems development.

### **What is a Meta-Model?**

We consider a systems development methodology to be a practice that offers an integration of a number of tools with a number of techniques for the application of these tools. Underpinning the tools is a “philosophy” (or a set of principles) which defends them by arguing that they realise certain qualities in a system developed and they facilitate or enhance the development process.

A meta-model of a systems development methodology is a high level model in which the activity prescribed is that of deciding on the most appropriate approach to adopt at the top level at specific points in the development process. Deciding on an approach could entail selecting a particular model for part of the development process [9].

A meta-model accepts that a system is in a state of evolution without presupposing a particular change pattern. That is important if methods are to be linked to solve a particular problem.

### **ED<sup>2</sup>, as a Meta-Model of BPR, is a Methodology**

In this paper, we combine the principles of Evolutionary Delivery, Evolutionary Development and BPR to render ED<sup>2</sup> as an approach to shaping and delivering an integrated business and systems strategy. ED<sup>2</sup> is a meta-model of both BPR and software development as it integrates the two into a company-wide effort to sustain elevated business performance.

ED<sup>2</sup> is not a straightforward combination of principles; it is rather the framework through which the system development process will benefit from a strategic pull from BPR. Under ED<sup>2</sup>, initiatives like Evolutionary Delivery

and Evolutionary Development will be enabled and successfully implemented. Further, BPR, when integrated with software development methodologies, will become a systematic approach to organisational transformation.

The reality is that BPR requires its own solution strategy in each situation. This is why Hammer proposes to start with a clean slate. What we need is a model that controls but is not prescriptive. ED<sup>2</sup> is such a model. It is a versatile and more flexible system development approach because it:

- exploits a full repertoire of known technical and managerial methods,
- provides a mechanism that enables a subset of these methods to be linked easily into an appropriate solution strategy for any given problem, and,
- can respond to new problems and new methods as they emerge.

ED<sup>2</sup> is called a meta-model because it incorporates and uses other models. It caters for system evolution, planning, process management and technical matters. The critical success factors of ED<sup>2</sup> are consistent with the ones of BPR.

### **What is the value of ED<sup>2</sup>?**

ED<sup>2</sup> would be required in all those circumstances where you have modern decentralised computing and related organisational transformation. It aims to develop a dynamic system, deliver it to real users and make them the point of reference when measuring the added value of the system (Figure 3).

The orientation of ED<sup>2</sup> is the customer. From the notion of delivering value to an external customer, we get the notion of business process. Within a business process, we have end-users who are the internal customers for software support. The software can be evaluated from an internal perspective (‘how

does it support the end user?') and an external perspective ('does using this software add value to the business process and its external customers?').

### CONCLUSION

ED<sup>2</sup> - as a meta-model of both business and systems development methodologies - is meeting our need for a systematic approach to organisational transformation and software development whilst largely preserving our investment in the traditional tools and techniques of software development methodologies.

### REFERENCES

[1] M. Pedler, J. Burgoyne, and T. Boydell, *The Learning Company: A Strategy for Sustainable Development*, McGraw-Hill, Berkshire, 1991. [2] M. Hammer, "Reengineering Work: Don't Automate,

Obliterate", *Harvard Business Review*, July-August, 1990. [3] D.E. Avison, and G. Fitzgerald, *Information Systems Development: Methodologies, Techniques and Tools*. 2nd ed., McGraw-Hill, Berkshire, 1995. [4] A. Kuryla, "Iterative Application Development: An Evolutionary Strategy for the 90's", Open Insight White Papers, 1997. [5] T. Gilb, and S. Finzi, *Principles of Software Engineering Management*. Addison-Wesley, Wokingham, 1988. [6] N. Williams, *Evolutionary Development* [On line], Available at: <http://siris.sund.ac.uk/~calnwi/html/swproto/evdev.html>, 1995. [7] M. Hammer, "Reengineering Work: Don't Automate, Obliterate". *Harvard Business Review*, July-August, 1990. [8] "Hammer Defends Reengineering", *The Economist*, 5 Nov 1994. [9] G.H. Galal and L. Johnson, *KBS Methodologies: Principles and Misconceptions*. Technical Report, Department of Computer Science, University of Brunel, Middlesex, 1993.