Business Process Re-engineering

Business process re-engineering (BPR) as commonly practised is based on a largely empirical approach. This generally means:

- Observing or otherwise identifying what people in the business area do;
- Analysing, categorising and sequencing the collected material using an input-process-output model;
- Identifying the core thread of value-added activity and sifting out non-value-added accretions.

One of the strengths of an approach like this is its grounding in reality – what people in the business operation actually do in delivering customer value.

The classic BPR approach may not however rigidly differentiate between *what* and *how*. A business process is seen as an ordered set of activities involving real people using real computer systems and other tools. Process improvements arise first from eliminating non-value-added steps, and then by increasing efficiency and standardisation where possible by system enhancement and/or replacement. A specification of business requirements based on a classic BPR approach may therefore not be 'technology-free'.

Another BPR feature is that other than the input-process-output model and criteria for determining what is and is not 'value-added', it does not necessarily impose rigid definitions on its analytical components ('process', 'subprocess' and 'task' or their equivalents). The decomposition is often pragmatic: analysis continues until a level is reached at which value-added and non-value-added components differentiate themselves.

As a generic application across a range of different industries the BPR approach has considerable merit. In administration-intensive contexts however it can be greatly strengthened by the sort of conceptual definitions and modelling principles which this book employs.

As is so often the case, it depends what you want to achieve. If the point is to make the best of current systems, to ensure that time, money and energy are not being wasted unnecessarily, but not to change systems to any extent, then a classic BPR approach will probably work, particularly one employing rigorous quantitative cost-benefit disciplines.

But wherever system changes are not ruled out from inception (which should be a fairly common context) classic BPR process flow analysis seems to exhibit a major flaw. The flaw is that it is not 'technology-free'.

A business process is typically seen as a sequence of actual activities people carry out to achieve a result:

> Receive form A; photocopy it and give a copy to department B; enter the data from form A into screen C of system D; …etc.

Even where 'workflow' is included, this would typically be included as just another enabling technology:

> Receive form A; scan it into workflow system E; enter the data from the scanned image of form A into screen C of system D; …etc.

Whatever systems feature in the business, the process is seen as something separate. The process is a bit like a postman's route, with the system components as letterboxes. Now and again the postman is able to dump all the mail for a number of people into one big letterbox serving the block of flats they all live in rather than having to go to every flat. From the postman's point of view this is 'automation'.

This book has a fundamentally different approach to process analysis. It doesn't deny that what people actually do (and what they may continue to do) is 'Receive form A; …' etc. But what it wants to get at is the *what* that has to be done, regardless of *how* it is done. A key part of the *what* is the logic, the business rules, the *why*. It is for this reason that the approach is particularly applicable to administration-intensive contexts like financial services, government and so on.

It is not so much to do with administration *per se*, but because of the kind of thing that administration is. Administration is very often something that can be treated abstractly without losing too much of its essence. Because of its very nature it can be translated into different formats (for example digitised), and so can its rules. A life assurance policy for example is, in the last analysis, a legal contract between a financial organization and another person or organization, in relation to one or more human lives. Almost everything about the contract and the process of setting it up can be treated abstractly, in a translatable (eg 'digitisable') fashion. It is not possible to treat for example the process of making a wooden chair in the same way. This is because it is of the essence of a chair that it is a concrete object. A chair cannot be translated into another form and stay a chair. A life assurance policy printed

on paper may be a physical object, but the contract itself only has to exist in that form if the rules say so.

We are in a metaphysical realm and we must head back. To do this by way of the postman's route, my concern with empirical BPR techniques is that without focusing on the fundamental *what* and *why* the best you might hope for is a perfect bicycle. But if you address the *what* and the *why*, the fundamental content and logic of recorded communication – storage, addressing, privacy, transportation, distribution etc – you could get email.

In order for a business process to exist as something people can point to and recognise, it has to be repeatable and repeated. A business process is generally implemented somehow in order to ensure it is repeatable. The implementation can take many forms: system components; written rules (for example about which system component should be used when and for what); rules and guidelines in people's heads; paper or electronic templates; electronic routing of links to stored digitised documents; and so on – and very often some or all of these in combination. If you depict the *what* and the *why* (eg if the amount of the claim is greater than £100 it must be authorised by an assessor) in the same diagram as the *how* (call up screen C of system D) and see the whole diagram as the 'claim process', you mix apples and pears and risk confusing what is necessary with what is contingent.

Regardless of what you are actually able to do (how much money you can spend, how much time you have, what systems you can change or replace), it should be possible to evaluate the implementation as a whole as one out of many possible ways of carrying out the essential content of the process. It is less easy to do this if you see the essential content and logic and the implementation as one thing which can be improved and tuned for efficiency and quality benefits.

Why should this matter?

It matters because there *is* a way of implementing the essential content and logic of a business process in a computer system such that you would never again have to see the business process as something separate. In the real world the option may not always be available on pragmatic grounds (cash, time, legacy). But when these constraints can be overcome, and particularly when they must be overcome, there really is a way to re-engineer a business.

It is time to pick up the main thread again.