BPR - sic transit gloria mundi
or
What else is new in management?

Kai A. Simon
kai@adb.gu.se

Göteborg University
School of Economics & Commercial Law
Department of Informatics

Abstract. Business Process Reengineering (BPR) has lost a lot of the attention it attracted during the recent years. This paper will briefly address possible reasons for this phenomenon, and then introduce one of the new current trends in the management field that recently have gained a considerable attention, namely CMM - the Capability Maturity Model.

Introduction

When looking at BPR - Business Process Reengineering today, using a life cycle approach and terminology, many people would say it has turned into a dog. Michael Hammer, has recently published a new book – ”Beyond Reengineering” – and Thomas Davenport, another well-known former reengineering proponent, wrote an article entitled ”Why Reengineering Failed”. Consulting companies have removed skills in reengineering from their recruiting advertisements, and it was actually a long time ago we could read a success story in the popular business press.

What are the lessons to learn? Has reengineering hit the wall, or are we entering a phase of consolidation, where euphoria makes place for reflection and superficiality turns into understanding of the real issues? If we believe Davenport, ”reengineering isn’t dead, it is effectively over”. But, when looking at corporate agendas, the struggle for improved quality, ever shorter cycle times, and cost reduction is far away from being over. The question to ask must therefore be, whether BPR has lost its position, and if so, why has this happened.

”The way is the goal”

the ancient chinese philosopher once said, and a lot of the reengineering discussion actually turned out to be concerned with just that: How to get from one organizational form to another, changing structural variables, and using information technology for improving communication

in the resulting work settings. In case of that sounding familiar, it used to be known as change or transition management. Many critics from academia and practice have claimed, that reengineering wasn't anything new, but a re-animation of tayloristic work principles, enhanced by IT-employment. If this statement holds true, why should so many companies and public organizations embark on it? Acting in a dynamic environment is hardly improved by an organization design that emphasises on cost and cycle-time reduction only, while maintaining close managerial control.

An alternative, more balanced answer could be that BPR in itself was not revolutionary, but that it resulted in an increased awareness of the fact that there are multiple aspects to be taken under consideration when initiating change projects – strategy, technology, processes, and people. This intention, to provide a more holistic approach, instead of sub-optimizing isolated elements of the organization, can be considered as the main contribution BPR has delivered to the world of change. In addition, BPR has a clear focus on what must be done, instead of what hasn't been done. The conclusion to draw might be that many companies have been over-emphasizing on specific aspects such as information technology, while disregarding the need of integrated change in all the dimensions under concern, and abandoning most of its inherited burden.

BPR in the press

We could find support for the claim for the end of BPR when considering the number of reengineering related publications in scientific and populistic publications. A review of the ABI/INFORM database, comprising around 1.000 journals and others periodic publications, revealed the following numbers of publications per annum.[1]

Year	# of publications
1988	9
1989	12
1990	18
1991	45
1992	145
1993	597
1994	1.047
1995	1.370
1996 (incl. 08/96)	589

Table 1: # of BPR related publications p.a. 1988-1996

[1] The database search was conducted with the search string "bpr *or* reengineering".

According to the death-proposition offered by many authors, the number of publications should be declining significantly with a beginning in 1995. As the figures above show, the number of publications has never as high as in this year. An extrapolation of the 1996 figure would result in a total number of 884 publications, which could indicate a trend break.

In the following, we will introduce an approach to organization development that has its origin in the software engineering field, but has become more and more fashionable as a concept for analyzing an organiztion's processes in a wider perspective, and may become a new "hot trend around".

Successor ahead?

Briefing the current trends in management, we can identify two topics that recently have gained a considerable amount of attention, *knowledge management* and the *Capability Maturity Model (CMM)*. While knowledge management focuses on how the competencies and skills of organizational members can be made accessible throughout the entire organization, the CMM was originally designed as a framework for software development processes, and has now, in the same way as reengineering[2], found another application area in the management field.

CMM - the Capability Maturity Model

The capability maturity model was developed by the Software Engineering Institute at Carnegie Mellon University, and describes five levels of maturity a company can achieve during their systems development initiatives, and the related management and change processes. In order to manage the different levels, a set of key process areas is defined, describing the aspects and mechanisms to take into consideration on each level.

The CMM was originally developed for the certification of government contractors on software development projects. However, its focus on process makes it appropriate for the analysis of reengineering-related improvements.

We can identify three changes associated with reengineering to which the CMM can apply: Processes, people skills, and management capabilities. Management is often reluctant to assess its maturity for a variety of reasons, one of which may be the fact that the assessment of management capabilities is a difficult task to undertake. The CMM can be used as a guideline with which management measures itself and the rest of the processes being performed in the organization. The model's basic suggestion is the necessity to understand the current state of an organization's processes before trying to leap forward with improvement approaches.

The importance of control is clearly emphasized during the maturation process. An organization should not expect major improvements in process performance if it cannot perform a process with the same degree of precision twice in a row. This prerequisition with its focus on

[2] The term *reengineering* was originally used as part of *software reengineering* several years before the term *BPR* was used, describing the process of reconsidering and resdesigning existing software applications.

the absence of variation in the organizational processes could allow a categorization of CMM in the TQM (Total Quality Management) domain, where a similar focus on the repeatability of processes is found.

The maturity model has another key implication. It suggests a fixed sequence from one level to the next. While it may be possible to reach the optimized stage within a very narrow domain, e.g. a manufacturing or product development process, the achievement of organization-wide improvements requires a methodological approach through the various stages.

Maturity Levels	Focus On	Key Software Process Areas (Institutionalized)	Results
Optimizing	*empowering individuals*	◆ Process change management ◆ Technology change management ◆ Defect prevention	Productivity & Quality
Managed	*empowering objects*	◆ Software quality management ◆ Quantitative process management	
Defined	*organization*	◆ Peer reviews ◆ Intergroup coordination ◆ Software product engineering ◆ Integrated software management ◆ Training program ◆ Organization process definition ◆ Organization process focus	
Repeatable	*individual project*	◆ Software configuration management ◆ Software quality assurance ◆ Software subcontract management ◆ Software project tracking and oversight ◆ Software project planning ◆ Requirements management	
Initial	*individual*	◆ None	Risk

Figure 1: Capability Maturity levels (MIL1)

As it was stated earlier, this model has its origin in the field of software engineering, but is now applied even on organizational change. In the following, let us investigate the five maturity levels and the key process areas being used in the model in two ways, one describing the application of the model on software engineering, the other one focusing on the model application on the domain of organizational change.

Level 1 - Initial

At this level, few of the organizational processes are defined or even identified as such. The efforts taking place are based uopn current work practices, and success depends primarily on individual efforts and skills. Project or process management is virtually conducted in an ad hoc

manner, and occasionally even chaotic, i.e. that there is no clear concept for how work shouldbe done and how coordination among different efforts takes place.

Aspect	Software engineering	Organizational change
General	• Lowest level of maturity • No formal software development processes or procedures • Unclear project scope • Reactive project management • No visibility, outside information upon request or at end of the project • Immature according to model	• Lowest level of maturity • No methodological approach used for process management • Unclear objectives / strategy • Reactive process management • No visibility, outside information unavailable • Immature according to model
Key process areas	None	None

Level 2 - Repeatable

At this level, fundamental project or process management mechanisms are established. They are used for tracking cost, the progress of the project, and functionality. The process are, if not clearly documented, at least established to an extent that allows to repeat earlier success in the conduction of similar projects.

From a business process management perspective, this level allows the company to track the state of the process with regard to cost and time, but effective performance requires uniform inputs to the process, requiring uniform handling. Aspects of flexibility aren't addressed.

Aspect	Software engineering	Organizational change
General	• Basic project management mechanisms - milestones, decision points, deliverables • Information available to outside at defined points	• Basic process management mehanisms - milestones, decision points, delivery points • Information available to outside at defined points
Key process areas	• *Software Project Planning*. Establishment of estimates for work to be done • *Software Project Tracking and Oversights*. Limited visibility into process, action can be taken in case of derivation from project plan • *Software Subcontract Management*.	• *Process Planning*. Existence of overall description of process and its steps, consideration of time / effort • *Process Monitoring and Oversights*. Limited visibility and ability to track derivations from plan, action can be taken in case of derivation. • *Choice of Consultant Assistance*.

	Selection and management of sub-contractors. Choice upon performance abilities, measurement against work plan • *Software Quality Assurance.* Audits and reviews of the development project, compliance with standards • *Software Configuration Management.* Maintenance of software integrity, based on identifiaction of software configuration	Selection and management of consultants. Choice upon performance abilities, measurement against work plan • *Quality Assurance.* Change team audits and reviews, compliance with process description and procedures • *Integrated change.* Maintenance of process integrity, based on initial process requirements and design

Level 3 - Defined

The processes for both management and operational activities are well documented, standardized, and integrated into a standard set of processes for the organization.Within projects the documented, approved versions of one or more of the organization's standard processes for operations and development are used.

At this level, there is a clear understanding of the processes being performed, their logic, and their organizational embedding. This includes the definition of workflows, the distribution of work among team members and different organizational units, and includes the identificaton of collaborative mechanisms for improved coordination. Even the issue of organizational and individual competencies is addresse through the identification of competence gaps and training and education needs.

Aspect	**Software engineering**	**Organizational change**
General	• Focus on organizational structure of software development team • Understandning of distribution of work and responsibility • Improved management capability of risk assessment and proactivity	• Understanding of organizational setting required to achieve common goals, often focus on self-managing teams • Understanding of activities and workflow • Capability of assessing risk for process derivation, proactive adjustment
Key process areas	• *Organization Process Focus.* Assign-ment of tasks and responsibilities to project members • *Organization Process Definition.*	• *Organization Process Focus.* Determi-nation of process team responsibilities, consideration of members core capabilities • *Organization Process Definition.*

	Definition of software processes, aiming at improved process performance. Improved visibility and process measurement	Definition of organizational processes within built-in improvement capability and monitoring mechanisms
	• *Training programs*. Provide team members with skills and knowledge. Includes identification of knowledge gaps and training needs	• *Training programs.* Provide process team members with skills and knowledge, identification of knowledge gaps and training needs
	• *Integrated Software Management*. Integration of software development and organizational processes according to organizational process assets	• *Integrated change management*. Integration of change process and organizational processes according to process performance capabilities
	• *Software Product Engineering*. Alignment of development processes with software engineering practices	• *Change management practice*. Alignment of change activities with change management principles and practices
	• *Intergroup Coordination.* Establishment of communication mechanisms between groups for coordination of activities and work practices	• *Process Coordination.* Establishment of coordination and communication mechanisms between different processes, development of IT-support for effective collaboration, e.g. workflow technology
	• *Peer review.* Identification and removal of software defects	• *Peer review.* Identification and removal of process pathologies for further improvement. Often performed in ”quality circle” manner

Level 4 - Managed

At this level, detailed measures of the organizational processes in terms of cost, cycle time and process and product quality are gathered. Both the product and the processes being performed to create it are quantitatively understood and controlled.

The measurement of multiple performance indicators at this level is used for proactively managing the processes towards improved efficiency, i.e. a focus is put on active control for keeping performance within the pre-defined control limits, and ensuring a variation-free process operation.

Aspect	Software engineering	Organizational change
General	• Progress and problem measurement	• Organizational processes are closely monitored for derivations and progress
	• Quantitative decision making	• Decision making upon quantitative data
	• Ability to predict outcomes	• Outcome determined upon premises and process repeatablity
	• Anticipation of risk before project start	• Identification and assessment of critical success factors for organizational processes before implementation
Key process areas	• *Quantitative Process Management.* Quantification of process performance, bounds identified by project members, process outside bounds is corrected	• *Process Management.* Identification of process scope and control limits. Exceeding control limits (often 3σ) means out of control and is corrected
	• *Software Quality Managment.* Definition of quantitative goals for software quality. Includes definition strategy and action plan	• *Process Performance Indicators.* Indicators such as time, cost, quality are etablished, including road-map for goal achievement

Level 5 - Optimizing

The final level is striving for establishing mechanisms for continuous process improvement (CPI) through the use of quantitative feedback from the process, and the piloting of innovations.

The measures being used can range from continuous improvement initiatives to major change efforts under reegineering banner. In addition, human resource aspects are taken under consideration for exploiting the innovative potential of organizational members. The major pre-requisition for achieving this level and possessing a high level of capability maturity is that the underlying organizational core processes are effectively managed according to the performance measures that have been set for them.

Aspect	Software engineering	Organizational change
General	• Explorative analysis of new development approaches	• Experimental, iterative design of new organizational processes for CPI and/or quantum leaps
	Revision and/or replacement of ineffective processes	• Iterative redesign of ineffectice processes through ongoing identification of current rocess pathologies

Key process areas	• Complete visibility into the process • *Defect prevention.* Identification of cause/eefect relationships for preventing occurence of defects, modification of processes where defects occur • *Technology Change Management.* Integration of new technologies into the development processes, includes identifiaction, selection and evaluation of emerging technologies • *Process Change Management.* Improvement of software development processes. Range from CPI to entirely new methods	• Complete monitoring of organizational performance measures • *Process Operations Management.* Identification of performance gaps and error sources. Iterative alignment of process with performance objectives • *Technology Change Management.* Exploration of new technology usable for process improvement, including pilot use of emerging technologies • *Making Change Constant.* Ongoing evaluation and change of organizational processes. Ranges from CPI to radical change of reengineering type

CMM, TQM and BPR

Even though the CMM originally stems from the need of the military to assess the capabilitites of software development firms bidding for tenders, it is now widely used for assessing and improving companies' capability of managing their business processes. The means used to achieve organizational improvement are result measuring and refinement. When looking at the steps being involved, the key process areas, we can find a striking similarity with the problem-solving approaches being used in Total Quality Management Programs. Indeed, many companies are using the CMM as an integrated part of their quality management efforts (PAUL 1992).

However, neither TQM nor the CMM provides guidance regarding the specific tools or technologies to be used for performing the processes under concern effectively. Software developers are not limited to the use of Object Oriented technology or specific programming languages, and when looking at business processes, there are no recommendations for using a certain way of IT-support, such as workflow technology, or process management approach. Instead, the model comprises a set of requirements that must be satisfied in order to enable the organization to handle their processes at certain levels of maturity.

Considering the realtionship between BPR and CMM, one could claim that it is similar to the relationship being found between quality management and reengineering, namely continuous improvement in small steps over a longer period of time, based on the existing processes, versus order of magnitude change striving for major improvements at a rapid pace.

However, CMM offers a set of requirements that must be fulfilled before an organization should consider to embark on reengineering projects. There have been proposals that a

company should at least have reached level 3 (Defined) before initiating BPR, since the necessary skills required for process understanding do not exist on the lower levels.

Conclusions

When looking at the life cycle of management ”trends”, we can observe an average life-time of about 5 years. Total Quality Management was introduced in the mid 80s, and lasted until the early 90s. At that point in time, to be more specific in 1990, Business Process Reengineering was introduced in the United States, and gained increasing attention until the mid 90s.

In this concern, it is interesting to note that many companies maintained their quality improvements programmes, or tried to integrate them with their BPR initiatives, e.g. the T50 programme at ABB was intentionally designed from a continuous improvement perspective, despite the fact that it has been used as an example for successful reengineering.

Considering the Capability Maturity Model as a general approach to process management and improvement, possessing characteristics from both TQM and BPR, the conclusion to draw is that many organizations now have reached a point where a business process approach is widely accepted. At the same time, they take a step back and do not solely embark on a single way to change, but use multiple approaches simultaneously in a situation dependent way, which opens for new ways of managing businesses and change.

Sources / References

MIL1; US Navy: Naval Computer and Telecommunications Station, New Orleans (NCTS NOLA), URL: http://www.nola.navy.mil/seicmm.html, Accessed: 961107

Paul, Jim (1992); Still room for Heroes, in: *Computerworld, Oct 12 1992, p 96*

Software Engineering Insitute, Carnegie Mellon University, URL: http://www.sei.cmu.edu, Accessed: 961108

Software Engineering Institute (1995); *The Capability Maturity Model: Guidelines for Improving the Software Process,* Addison-Wesley Publishing Company