

8-16-1996

PROCESS REENGINEERING AND THE DATA FLOW DIAGRAM (DFD)

Lerle P. Martin
California State University, Sacramento

Follow this and additional works at: <http://aisel.aisnet.org/amcis1996>

Recommended Citation

Martin, Lerle P., "PROCESS REENGINEERING AND THE DATA FLOW DIAGRAM (DFD)" (1996). *AMCIS 1996 Proceedings*.
234.
<http://aisel.aisnet.org/amcis1996/234>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 1996 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

PROCESS REENGINEERING AND THE DATA FLOW DIAGRAM (DFD)

Merle P. Martin
California State University, Sacramento

INTRODUCTION

We no longer can be content with automating what is there, but must concentrate harder on changing basic system components and flows. We must influence rather than adorn our information systems. We can describe this more proactive approach as *Process Reengineering*.

The purpose of this paper is to demonstrate the power of process reengineering thinking *within the limited confines of the typical systems analyst*. For our purposes, we will define the systems analyst's confines as the Data Flow Diagram (DFD) used in the Analysis Phase of the Systems Development Life Cycle (SDLC).

PROCESS REENGINEERING

APPROACHES

Reengineering can be defined as reshaping the organization

... to take advantage of technological advances, make better use of information, and improve the firm's ... productivity. (Stair, 1992, p. 470)

In this context, software reengineering is a subset of the larger set of objectives referred to as Business Process Reengineering (BPR) (Pfrenzinger, 1992).

For the purposes of this paper, *process reengineering* is reshaping the firm's information processes in order to improve overall productivity. We require two elements to operationalize this definition: (a) a set of process reengineering rules and (b) a venue (playing field) on which the information systems analyst can apply these rules. The venue we have selected is the DFD. Our process reengineering rules will be derived by investigating the intersection of two global views of process reengineering

--Total Quality Management (TQM) and Organizational Reengineering.

TOTAL QUALITY MANAGEMENT

TQM is

... the application of quality principles for the integration of all functions and processes of the organization. (Ross, 1993, p. 325) Harrington (1992) describes 12 ways of streamlining processes. Martin (1995) narrows this list to 6 principles for reengineering *information* system processes. These rules are: (1) eliminate duplicate processes, (2) eliminate processes that are not value added, (3) simplify processes, (4) reduce process cycle time, (5) simplify language, and (6) standardize processes.

ORGANIZATIONAL REENGINEERING

Hammer and Champy (1994, p. 32) define reengineering in an organizational sense as:

the fundamental rethinking and radical redesign of business processes to achieve dramatic improvement in critical, contemporary measures of performance, such as cost, quality, service, and speed.

Hammer and Champy suggest paradigm bursting approaches often beyond the authority and capabilities of the typical information systems analyst. We can glean, nevertheless, some key principles which can be applied in the analyst's restricted arena. These are: (1) combine several jobs into one (functional cohesion), (2) allow workers to make decisions, (3) have process steps performed in a natural order, (4) design multiple versions of complex processes, (5) have work performed where it makes sense, (6) reduce checks and controls, (7) minimize reconciliation, (8) establish a single point of customer contact (case manager), and (9) consider hybrid centralized/ decentralized systems.

Clearly some of these organizational reengineering principles duplicate or subsume the TQM principles. A set of operationalized process reengineering rules will combine these two sets. First, however, we must provide to the information systems analyst a venue for applying this rule set. This venue is the Data Flow Diagram (DFD).

DATA FLOW DIAGRAM

In a structured design environment, DFD use proceeds in the following sequence (Yourdon, 1988): (1) hierarchical chart of system processes, (2) physical DFDs of the current system (both the *Whats* and *How's* of the system), (3) logical DFDs of the current system (stripping away the *How's*), (4) logical DFDs of the proposed system (modifying the *Whats*), and (5) physical DFDs of the proposed system (adding back in new *How's*).

Our process reengineering target is the transition between the logical DFDs of the current and proposed systems (Steps 3 and 4 above). Here we can change what is happening (the system foundation) without being bothered by implementation details (the who, when, where, and how).

REENGINEERING OPPORTUNITIES

Exhibit 1 identifies an example of 6 process reengineering opportunities. The key to the table is that it can be sorted by opportunity, by benefit, or by symptom--depending on specific analyst needs. This exhibit has for its last column a narrative description of what to look for on the DFD. This can be done graphically as shown in the next section of this paper. We have developed a total of 14 such rules so far.

DFD REENGINEERING EXAMPLES

One example will be illustrated, deriving from organizational reengineering principles; this is the Customer Focus (use of a Case Manager). Customers often become frustrated by having to deal with multiple points (offices, persons, etc.) in a system. No one point takes responsibility for the entire product process. One solution is to establish a Case Manager who (a) coordinates all processes and (b) is the single point of contact between the customer and the system. The DFD is transformed from a single-level left to right entity, process, process, entity flow to a multi-level flow where each process interfaces with a Case Manager.

SUMMARY

We must change our emphasis to making meaningful changes to information systems, rather than merely adorning them with the latest technology. To do this, we need to use process redefining methodologies. One means of bringing these broad methodologies into the specific, often organizationally narrow, scope of Systems Analysis is by use of the Data Flow Diagram. We have shown opportunities for significantly transferring the logical DFD of the current system to that of the proposed system.

Finally, it is recognized that we may be building potential frustration by leading typical systems analysts to recommend reengineering changes that may exceed the organization the scope of their responsibilities. We should, however, prefer to err in this direction rather than continue to allow systems analysts to recommend syntactic, software changes which have meager impact on total firm effectiveness (Moad, 1994).

References available upon request.

EXHIBIT 1: REENGINEERING OPPORTUNITIES

<i>Reengineering Opportunity</i>	<i>How Implemented</i>	<i>Benefits</i>	<i>What Symptoms To Look For On The DFD</i>
Line-Balancing	Make Simple, Complex Process Into Multiple Processes	Reduce Bottlenecks Reduce System Time Variance	Process With Multiple Output Dataflows
Outsourcing	Allocate Processes To Vendors/Customers (External Entities)	Reduce Costs Reduce System Time Reduce System Complexity	Processes That Can Be Done By External Entities
Value-Added	Eliminate Processes That Do Not Add Value To System	Reduce Costs Reduce System Time Reduce System Complexity	Processes That Don't Change Information Flow
Seamless Database	Establish One Integrated, Distributed Database	Reduce Costs Reduce Redundancy Eliminate Issue Of Centralized/ Decentralized Control	Multiple Data Stores
Customer Focus	Establish Single Point Of Customer Contact (Case Manager)	Reduce Customer Dissatisfaction Adopt More Quickly To Changing Customer Needs Pinpoint Faulty Processes	Multiple Processes Linking Source And Sink To External Entities Of Same Name
Buffering	Connect Processes Through Data Store That Acts As Workload Buffer	Reduce Reliance Of One Process On Another Reduce System Impact Caused By Process Variability	Several Processes That Could Be Connected Through Intermediate Data Store