SYSTEMS REENGINEERING

Industrial, organizational, and enterprise responsiveness to continuing challenges is very clearly a critical need today. One of these challenges is change of all sorts. This is accomplished by continually providing products and services of demonstrable value to customers. To do this requires efficiently and effectively employing leadership and empowering employees such that systems engineering and management strategies, organizational processes, human resources, and appropriate technologies are each brought to bear to produce high-quality, trustworthy, and sustainable products and services. There is an ongoing need for continual revitalization in the way organizations and enterprises do things, so that they are always done better. This would be true even if the external environment were static and unchanging. However, in a period of high-velocity changes, continual change and associated change in processes and products must be considered a fundamental rule of the game for progress.

Change has become a very popular word today in management and in technology. There are a variety of change models and change theories. Some seek to change to survive. Others seek to change to retain competitive advantage. This article examines change in the form of reengineering. There are a variety of names given to the number of change-related terms now in use: reengineering, restructuring, downsizing, rightsizing, redesign, enterprise transformation, and many others. Reengineering is probably the most often used word, and systems reengineering is the title chosen here. There are many approaches to reengineering, and some of these are briefly examined here. Expansion of these discussions may be found in Refs. 1 and 2.

Figure 1 represents a generic view of reengineering. Reengineering can be discussed from several perspectives: from the structural, functional, and purposeful aspects of reengineering or at the level of systems management, process, or product. Reengineering issues may be examined at any, or all, of the three fundamental systems engineering life cycles: research, development, test, and evaluation (*RDT&E*); systems acquisition, procurement, or production; or systems planning and marketing; all discussed in Refs. 1 and 2, on which this article is based. Within each of these life cycles, reengineering can be considered at any or all of the three generic phases of definition, development, or deployment. The level of systems management examines the enterprise as a whole and considers all organizational processes within the organization for improvement through change. At the level of process reengineering, only a single process is redesigned, with no fundamental or radical changes in the structure or purpose of the organization as a whole. When changes occur, they may be radical and revolutionary or incremental and evolutionary at the level of systems management, process, product, or any combination. The scale of improvement efforts may vary from incremental and continuous improvement, such as generally advocated by quality management efforts, to radical change efforts that affect organizational strategy and scope, and systems management itself.

One fundamental notion of reengineering, however, is that it must be directed top down if it is to achieve significant and long-lasting effects. Thus, there should be a strong, purposeful, systems management orientation to reengineering, even though it may have major implications for such lower-level concerns as the structural facets of a particular product. A major objective of reengineering is to enhance the performance of a legacy system or a legacy product or service. Thus, reengineering may support a variety of other desirable objectives, such as better integration of a product with other products and improved maintainability.

This article is organized as follows. First, some definitions of reengineering are provided. Then, some of the many perspectives that have been taken relative to reengineering are viewed at the levels of

- product
- process or product line
- systems management

PRODUCT REENGINEERING

The term *reengineering* could be used to mean a reworking or retrofit of an existing product. This could be interpreted as maintenance or refurbishment. Or reengineering could be interpreted as reverse engineering, in which the characteristics of an already engineered product are identified, so that the original product can be subsequently modified and reused or so that a new product with the same purpose and functions may be obtained through a forward engineering process. Generally the term product can also refer to service and we can reengineer at the level of products and/or services. Inherent in these notions are two major facets of reengineering:

1. Reengineering improves the product or system delivered to the user in enhanced reliability or maintainability or for an evolving user need.
2. Reengineering increases understanding of the system or product itself.

This interpretation of reengineering is almost totally product focused.

Product reengineering is the examination, study, capture, and modification of the internal mechanisms or function of an existing product to reconstitute it in a new form that has new functional and nonfunctional features, often to take advantage of newly emerged technologies, but without major change in the inherent purpose of the product. This definition indicates that product reengineering is basically structural reengineering with, at most, minor changes in purpose and function of the product that is reengineered. This reengineered product could be integrated with other products that have function rather different from that in the initial deployment. Thus, reengineered products could be used, together with this augmentation, to provide new functions and serve new purposes. A number of synonyms for product reengineering easily come to mind: among these are renewal, refurbishing, rework, re-

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 2007 John Wiley & Sons, Inc.

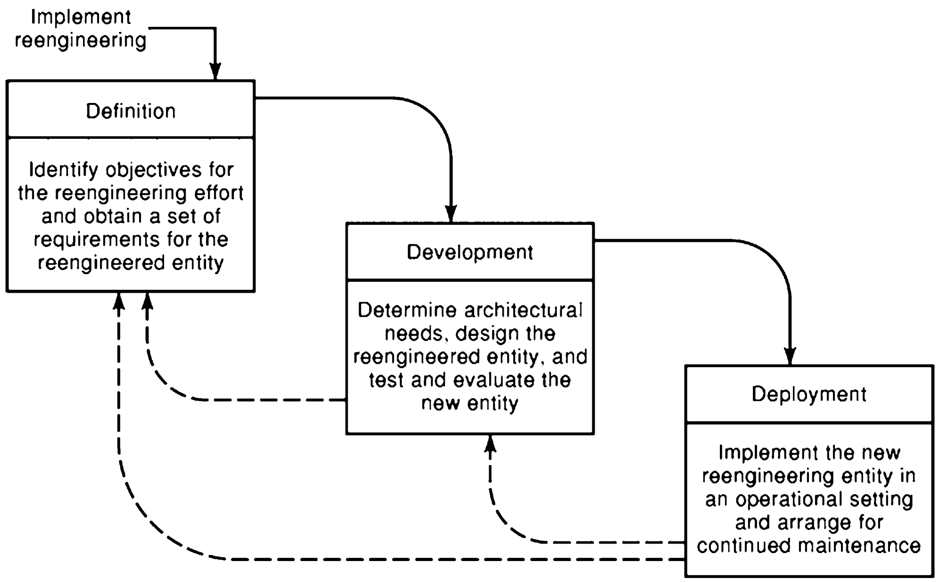

Figure 1. Generic implementation of reengineering at the level of product, process, or systems management.

pair, maintenance, modernization, reuse, redevelopment, and retrofit.

A specific example of a product reengineering effort is taking a legacy system written in COBOL or FORTRAN, reverse engineering it to determine the system definition, and then reengineering it in C++ or some other high-level language. Depending on whether any modified user requirements are to be incorporated into the reengineered product, it would be forward engineered after the initial development (technical) system specifications or determine user requirements and user specifications had been determined, and these would be updated. This reverse engineering concept (3), in which salient aspects of user requirements or technological specifications are recovered by examining characteristics of the product predates the term product reengineering and occurs before the forward engineering that comprised the latter portions of product reengineering.

Figure 2 illustrates product reengineering conceptually. An IEEE software standards reference (4) states that "reengineering is a complete process that encompasses an analysis of existing applications, restructuring, reverse, and forward engineering." The IEEE Standard for Software Maintenance (5) suggests that reengineering is a subset of software engineering composed of reverse engineering and forward engineering. We do not disagree at all with the definition, but prefer to call it product reengineering for the reasons just stated. There are two other very important forms of reengineering, and it is necessary to consider reengineering at the levels of processes and systems management to take full advantages of the major opportunities offered by generic reengineering concepts. Thus, the qualifier *product* appears appropriate in this context.

Reengineering at the product level has received much attention in recent times, especially in information technology and software engineering areas. This is not a subject that is truly independent of reengineering at the levels of either systems management or of a single life cycle process. It is also, related to notions of systems integration. Product reengineering is generally needed whenever development of an entirely new product is too expensive, when there is no suitable and available commercial product, and when the current system does not fulfill some of the functional requirements or such nonfunctional requirements as trustworthiness.

Much product reengineering is closely associated with reverse engineering to recover either design specifications or user requirements, followed by refining these requirements or specifications and forward engineering to create an improved product. The term reverse engineering, rather than reengineering, was used in one of the early seminal papers in this area (6) concerned with software product reengineering. In this work and in a related chapter on the subject (7), the following activities represent both the taxonomy of and phases for is denoted here as *product reengineering:*

1. *Forward engineering* is the original process of defining, developing, and deploying a product or realizing a system concept as a product.
2. *Reverse engineering,* sometimes called inverse engineering, is the process through which a given system or product is examined to identify or specify the definition of the product at the level either of technological design specifications or of system or user requirements.
    a. *Redocumentation* is a subset of reverse engineering in which a representation of the subject system or product is re-created to generate functional explanations of original system behavior and, perhaps more important, to aid the reverse engineering team in better understanding the system at a functional and structural level. There are a number of redocumentation tools available for software, and some of these are cited in these works. One of the major purposes of redocumentation is to produce new documentation for an existing product whose existing documentation

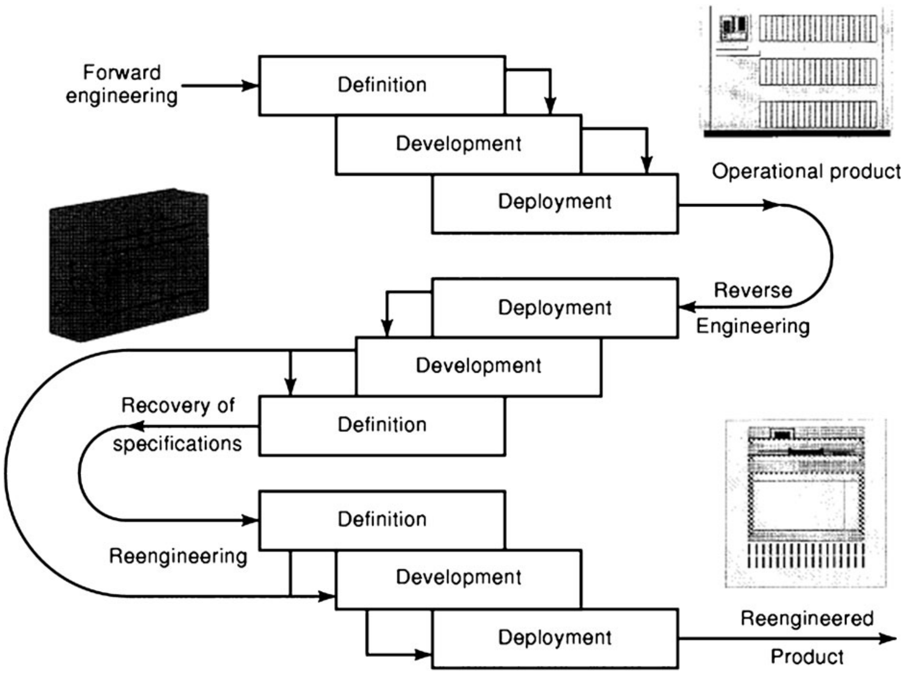

Figure 2. Basic notions of product reengineering as a sequence of forward, reverse, forward engineering.

is faulty and perhaps virtually absent.

b. *Design Recovery* is a subset of reverse engineering in which the redocumentation knowledge is combined with other efforts, often involving the personal experiences and knowledge of others about the system, that lead to functional abstractions and enhanced product or system understanding at the level of function, structure, and even purpose. We prefer to call this deployment recovery, development recovery (which includes design recovery), and definition recovery, depending on the phase in the reverse engineering life cycle at which the recovery knowledge is obtained.

3. *Restructuring* involves transforming the information about the original system structure into another representational form. This generally preserves the initial functions of the original system or slightly modifies them purposefully in accord with changes in user requirements for the reengineered system. The terms deployment restructuring, development restructuring, and definition restructuring are appropriate disaggregations of the restructuring notion.
4. As defined here, *reengineering* is equivalent to redevelopment engineering, renovation engineering, and reclamation engineering. Thus, it is closely related to maintenance and reuse. Product reengineering is the re-creation of the original system in a new form that has improved structure but generally not much altered purpose and function. The nonfunctional aspects of the new system may differ considerably from those of the original system, especially with respect to quality and reliability.

Figure 2, which illustrates product reengineering, involves these activities.

We can recast this by considering a single phase for definition, development, and deployment that is exercised three times. Then we see that there is a need for recovery, redocumentation, and restructuring as a result of the reverse engineering product obtained at each of the three phases. This leads us to suggest Fig. 3 as an alternative way to represent Fig. 2 and as our interpretation of the representations generally used for product reengineering. Many discussions, such as those just referenced, use a three-phase generic life cycle of requirements, design, and implementation. Implementation would generally contain some of the detailed design and production efforts of our development phase and potentially less of the maintenance efforts that follow initial fielding of the system. The restructuring effort, based on recovery and redocumentation knowledge obtained in reverse engineering, is used to effect deployment restructuring, development restructuring, and definition restructuring. To these restructured products, which might be considered reusable products, we augment the knowledge and results obtained by detailed consideration of potentially augmented requirements. These augmented requirements are translated, together with the results of the restructuring efforts, into the outputs of the reengineering effort at the various phases to result ultimately in the reengineered product.

There are a number of objectives in, potential uses for, and characteristics of product reengineering, which are neither mutually exclusive nor collectively exhaustive and include the following (8, 9):

1. Reengineering may help reduce an organization's risk of product evolution through what effectively amounts to reuse of proven subproducts.
2. Reengineering may help an organization recoup its product development expenses by constructing new products based on existing products.

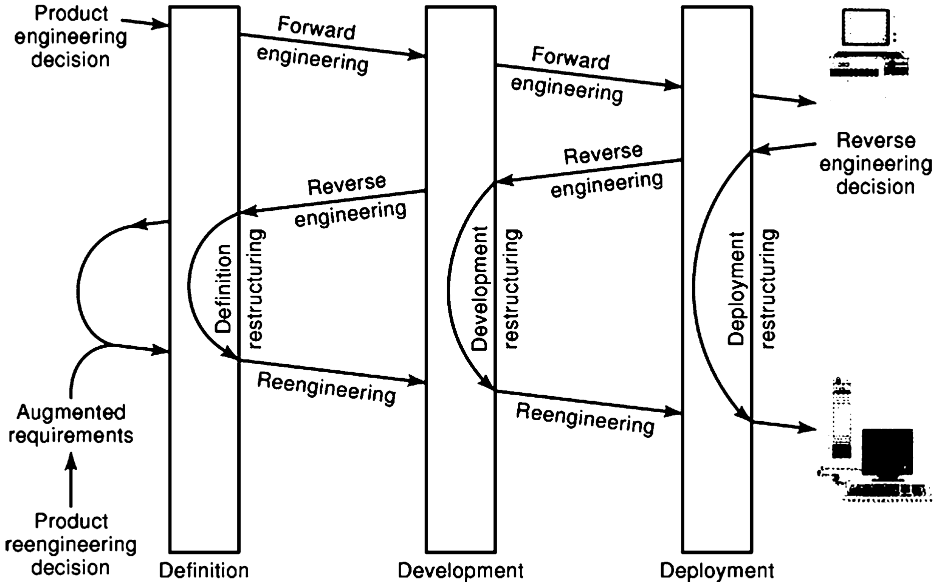

Figure 3. Expanded notion of product reengineering.

3. Reengineering may make products easier to modify to accommodate evolving customer needs.
4. Reengineering may make it possible to move the product, especially a software product, to a less expensive operational environment, such as from COBOL to an object-oriented language or from a mainframe to a server.
5. Reengineering may be a catalyst for automating and improving product maintenance, especially by obtaining smaller subsystems with better defined interfaces.
6. Reengineering a product may result in a product with much greater reliability.
7. Reengineering may be a catalyst for applying new technologies, such as CASE tools and artificial intelligence.
8. Reengineering may prepare a reengineered product for functional enhancement.
9. Reengineering is big business, especially considering the major investment in legacy systems that need to be updated, maintained, and improved in functions.

In short, reengineering provides a mechanism that enhances an understanding of systems so that this knowledge can be applied to produce new and better systems and products.

Planning for product reengineering is essential, just as it is for other engineering efforts. Product engineering planning involves the standard systems engineering phases (1–10):

1. *Definition Phase.*

   a. Formulation of the reengineering issue to determine the need for and requirements to be satisfied by the reengineered product, and identification of potential alternative candidates for reengineering.

   b. Analysis of the alternatives to enable determining costs and benefits of the various alternatives.

   c. Interpretation and selection of a preferred plan for reengineering.

2. *Development Phase,* in which the detailed specifications for implementing the reengineering plan are determined.
3. *Deployment Phase,* in which operational plans, including contracting, are set forth to enable reengineering the product in a cost-effective and trustworthy manner.

Although reengineering has proven to be a successful way to improve product systems, it requires a demonstration that there will be benefits associated with the effort that justify the costs. Usually, it is necessary to compare the costs and benefits of reengineering a product with developing an entirely new one.

Unfortunately, it is generally not easy to estimate the cost of a reengineering effort or the benefits that will follow from it. In some cases, this is easier for a reengineered product than for a totally new product because the existing legacy product often provides a baseline for these estimations. Sneed (11) suggests 16 relevant attributes for such an analysis; they are listed in Table 1.

A number of authors have suggested specific life cycles that lead to a decision whether to reengineer a product and, in support of a positive decision, support a product reengineering life cycles (12, 13). The following are some of the accomplishments needed.

1. Initially, there exists a need to formulate, assess, and implement definitional issues associated with the technical and organizational environment. These issues include organizational needs relative to the area under consideration and the extent to which technology and the product or system under reengineering consideration supports these organizational needs.

2. Identification and evaluation of options for continued development and maintenance of the product under consideration, including options for potentially outsourcing this activity.
3. Formulation and evaluation of options for the composition of the reengineering team, including insourcing and outsourcing possibilities.
4. Identification and selection of a program of systematic measurements that determine the cost efficiency of the identified reengineering options and facilitate selection of a set of options.
5. The legacy systems in the organizations need to be examined to determine the extent to which these existing systems are currently functionally useless and in need of total replacement, functionally useful but with functional and nonfunctional defects that could be remedied through product reengineering, or fully appropriate for the current and intended future uses.
6. A suite of tools and methods to allow reengineering needs to be established. Method and tool analysis and integration are needed to provide for multiple views across various abstraction levels (procedural, pseudoprocedural, and nonprocedural) encountered in reengineering.
7. A process for product reengineering needs to be created on the basis of the results of these earlier steps that provides for reengineering complete products reengineering systems, and for incremental reengineering efforts that are phased in over time.
8. Major provisions for education and training must be made.

   This is more of a checklist of needed requirements for a reengineering process than a specification of a life cycle for the process itself. Through perusal of this checklist, we should be able to establish an appropriate process for reengineering in the form of Figs. 2 or 3.

This article does not describe the large number of tools available to assist and support the product reengineering process. These vary considerably depending on the type of product reengineered. A number of tools for software reengineering are described in Muller (14) and the bibliography in this article.

There are several needs that must be considered if a product reengineering process is to yield useful results:

1. A need to consider long-range organizational and technological issues in developing a product reengineering strategy.
2. A need to consider human, leadership, and cultural issues, and how these will be affected by the development and deployment of a reengineered product.
3. It must be possible to demonstrate that the reengineering process and product are cost effective and of high quality and that they support continued evolution of future capabilities.
4. Reengineered products must be considered within a larger framework that also considers the potential need for reengineering at the levels of systems management and organizational processes because it is generally a mistake to assume that technological fixes can resolve organizational difficulties at these levels.
5. Product reengineering for improved postdeployment maintainability must consider maintainability at the process level rather than at the product level only, such as would result from rewriting source code statements. Use of model-based management systems or code generators should yield much greater productivity in this connection than rewriting source code.
6. Product reengineering must consider the need for reintegrating the reengineered product with legacy systems that have not been reengineered.
7. Product reengineering should increase conformance to standards as a result of the reengineering process.
8. Product reengineering must consider legal issues associated with reverse engineering.

The importance of most of these issues is relatively self-evident. Issues surrounding legality are in a state of flux in product reengineering, in much the same way as they are for benchmarking. They deserve special commentary here.

It is clearly legal for an organization to reverse engineer a product that it owns. Also, little debate exists at this time as to whether it is legal to infer purpose from the analysis of a product without any attempt to examine its architectural structure or detailed components and then to recapture that purpose through a new development effort (the so-called black-box approach). Major questions, however, surround the legality of white-box reverse engineering, in which the detailed architectural structure and components of a system, including software, are examined to reverse engineer and reengineer it. The major difficulty stems from the fair-use provisions in copyright law and the fact that fair-use provisions differ from those associated with the use of trade secrets for illicit gain. Copyrighted material cannot be secret because copyright law requires open disclosure of the copyrighted material. Because software is copyrighted, not patented, trade secret restrictions do not apply. There is a pragmatic group that says white-box reengineering is legal, and a constructionist group that says it is illegal (15, 16). Those who suggest that it is illegal argue that obtaining trade secrets is not illegal, but the subsequent use of these for illicit gain. These issues will continue to be the subject of much debate. Many of the ethical issues in product engineering are similar to those in benchmarking and other approaches to process reengineering.

Some useful guidelines applicable primarily to product reengineering follows:

1. Reverse engineering procedures can be performed only on products that are the property of the reengineering organization or that have come into its possession legally.
2. No patent exists that would be infringed by a functional clone of a computer program, and no one can

be under a contractual obligation not to reverse engineer the original product.

3. A justifiable procedure for reverse engineering is to apply an input signal to the system or product being reengineered, observe the operation of the product in response to these inputs, and characterize the product functionally based on operation. Then an original product (computer code in the case of software) should be written to achieve the functional characteristics that have been observed.

In the case of computer programs, it is permissible to disassemble programs available in object code form to understand the functional characteristics of the programs. Disassembly is used only to discover how the program operates, and it may be used only for this purpose. The functional operating characteristics of the disassembled computer program may be obtained, but original computer code should be prepared from these functional characteristics. This new code must serve this functional purpose.

Reengineering is accompanied by a variety of risks associated with processes, people, tools, strategies, and the application area. These can be managed through risk management methodologies (1–17). These risks derive from a variety of factors:

- Integration risk that a reengineered product cannot be satisfactorily integrated with legacy systems.
- Maintenance improvement risk that the reengineered product will exacerbate, rather than ameliorate, maintenance difficulties.
- Systems management risk that the reengineered product attempts to impose a technological fix on a situation whose major difficulties derive from problems at the level of systems management.
- Process risk that a reengineered product that represents an improvement in a situation where the specific organizational process is to be used is defective.
- Cost risk that major cost overruns are required to obtain a reengineered product that meets specifications.
- Schedule risk that delays are encountered to obtain a deployed reengineered product that meets specifications.
- Human acceptance risk of obtaining a reengineered product that is not suitable for human interaction or is unacceptable to the user organization for other reasons.
- Application supportability risk that the reengineered product does not really support its intended application or purpose.
- Tool and method availability risk of proceeding with product reengineering based on promises for a method or tool needed to complete the effort that does not become available or that is faulty.
- Leadership, strategy, and culture risks arising from imposing a technological fix in the form of a reengineered product on an organizational environment that cannot adapt to the reengineered product.

Clearly, these risks are not mutually exclusive, the risk attributes are not independent, and the listing is incomplete. For example, legal risks could be included.

There is clearly a very close relationship between product reengineering and product reuse. The reengineering of legacy software and the reuse-based production of new software are closely related concepts. Often, the cost of developing software for one or a few applications is almost the same as the cost of developing domain reuse components and reengineering approaches to legacy software. Ahrens and Prywes (18) describe some of these relationships in an insightful work.

PROCESS REENGINEERING

Reengineering can also be instituted at the process and systems management levels. At the level of processes only, the effort would be almost totally internal. It would consist of modifications to existing life cycle processes to better accommodate new and emerging technologies or new customer requirements for a system. For example, an explicit risk management capability might be incorporated at several different phases of a given life cycle and accommodated by a revised management process. This could be implemented into the processes for RDT&E, acquisition, and systems planning and marketing. Basically, reengineering at the level of processes consists of determining or synthesizing an efficacious process for fielding a product based on knowledge of generic customer requirements and the objectives and critical capabilities of the systems engineering organization. Figure 4 illustrates some of the facets of process reengineering. Process reengineering may be instituted to obtain better products or a better organization. There are three ways for attempting process improvement:

- new process development,
- process redevelopment, or process reengineering, or
- continuous process improvement over time.

New process development is necessary because of a strategic level change, such as when a previously outsourced development effort is insourced and there is no present process on which to base the new one. Benchmarking, discussed here, is one way of accomplishing this. Process redevelopment, or reengineering, should be implement when the existing process is dysfunctional or when the organization wishes to keep abreast of changing technology or changing customer requirements. Continuous process improvement is less radical and can be carried out incrementally over time. Each of these involves leadership, strategy, and a team to accomplish the effort.

In accordance with this discussion and analogous to our definition of product reengineering, we offer the following definition. *Process reengineering is the examination, study, capture, and modification of the internal mechanisms or functions of an existing process or systems engineering life cycle to reconstitute it in a new form with new functional and nonfunctional features, often to take advantage of newly emerged or desired organizational or technological capabilities, but without changing the inherent*

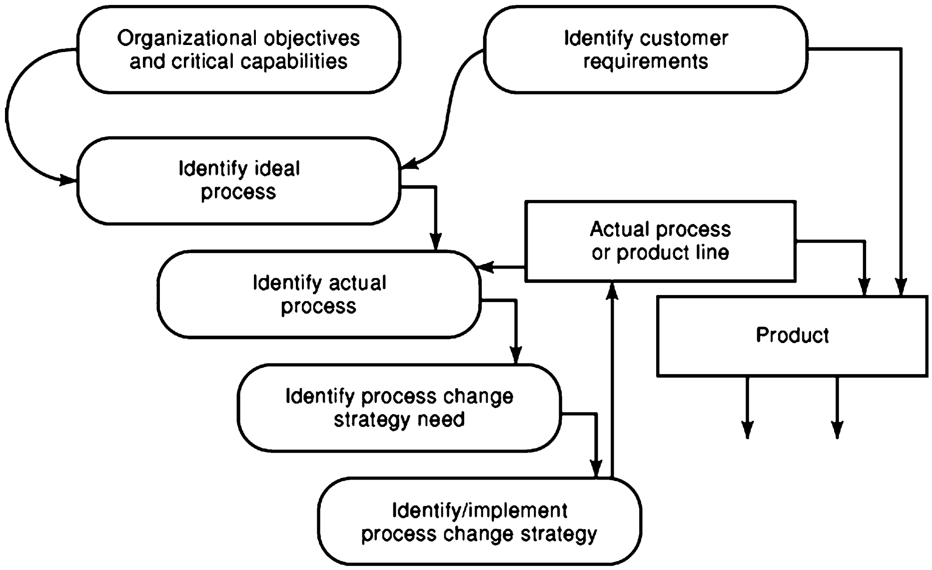

Figure 4. Conceptual illustration of process reengineering.

purpose of the process itself.

Concurrent Engineering

Often, it is desired to produce and field a system relatively rapidly. The life cycle processes needed to achieve this could be accelerated if it were possible to accomplish phases of the relevant life cycles more or less concurrently. Concurrent engineering is a systems engineering approach to the integrated coincident design and development of products, systems, and processes (19–22).

The basic tasks in concurrent engineering are much the same as the basic tasks in systems engineering and management. The first step is that of determining customer requirements; these are then translated into a set of technical specifications. The next phase involves program planning to develop a product. Often, especially in current engineering, this involves examining the current process and generally refining existing processes to deliver a quality product that meets both customer needs and cost and schedule requirements.

In concurrent engineering, the very early and effective configuration of the systems life cycle process takes on special significance because the simultaneous development efforts need to be carefully coordinated and managed to forestall significant increases in cost and product time or significant deterioration in product quality. The use of coordinated product design teams, improved design approaches, and stringent standards are among the aids that can enhance concurrent engineering efforts.

Achieving a controlled environment in concurrent engineering and system integration requires the following:

1. *Information integration and management.* It must be possible to access information of all types easily and to share design information across the levels of concurrent design effectively and with control. Design information, dependencies, and alterations must be tracked effectively. The entire configuration of the concurrent life cycle process must be effectively monitored and managed.
2. *Data and tool integration and management.* It must be possible to integrate and manage tools and data so that there is interoperability of hardware and software across several layers of concurrency.
3. *Environment and framework integration,* or *total systems engineering.* It must be possible to ensure that process is directed at evolution of a high-quality product and that this product is directed to meet the needs of the user in a trustworthy manner that is endorsed by the customer. This requires integrating the environment and framework, or the processes, for the systems engineering and management efforts.

There is a close relationship between concurrent engineering and systems integration; Andrews and Leventhal (23), Kronloff (24), and Schefstrom and van den Broek (25) provide several details concerning the method, tool, and environment integration needed to implement concurrent engineering and other systems engineering efforts.

Compression of the life cycle phases that occurs in concurrent engineering poses more of a problem. The macroenhancement approaches to systems engineering, especially software systems engineering (26), are particularly useful in this regard. These include prototyping for system development, use of reusable systems and subsystems, and expert systems and automated program generation. Use of these can compress the overall time needed by parallel subsystem life cycles in a manner that is compatible with the engineering of a trustworthy product or service.

Formally, very little is new in the subject of concurrent engineering. Development phases are simply accelerated through their concurrent implementation, at least on the surface. Concurrent engineering, however, places a much greater reliance on strategic planning and systems management and requires greater attention to processes to ensure that they are well deployed and to the resulting integration to ensure success.

Carter and Baker (27) indicate that success in concurrent engineering depends very much on maintaining a proper balance between four important dimensions:

- Organizational culture and leadership and the necessary roles for product development teams.
- Communications infrastructure for empowered multidisciplinary teams.
- Careful identification of all functional and nonfunctional customer requirements, including those product and process facets that affect customer satisfaction.
- Integrated process and product development.

They identify approaches at the levels of task, project, program, and enterprise to enable realizing the proper environment for concurrent engineering across each of these four dimensions. Each of the four dimensions has a number of critical factors, and these may be approached at any or all of the levels suggested. A matrix is suggested to enable identifying the needed development areas to ensure definition, development, and deployment of an appropriate concurrent engineering process and process environment. The reference cited provides a wealth of pragmatic details for determining concurrent engineering process needs.

It is also noted (27) that five major roadblocks often impede development of a concurrent engineering process environment:

1. The currently available tools are not adequate for the new environment.
2. There are a plethora of noninteroperable computers, networks, interfaces, operating systems, and software in the organization.
3. There is a need for appropriate data and information management across the organization.
4. Needed information is not communicated across horizontal levels in the organization.
5. When correct decisions are made, they are not made in a timely manner.

Approaches are suggested to remove each of these roadblocks to enable developing a concurrent engineering process. Presumably this needs to be implemented continuously, as appropriate for a given organization, rather than attempting a revolutionary change in organizational behavior. Several worthwhile suggestions for implementation are provided.

Integrated Product and Process Development

In many ways, integrated product development (*IPD*) is an extension of concurrent engineering. In a work that focuses on the importance of requirements management, Fiksel (28) states that concurrent engineering is more accurately called integrated product development. It is also closely related to the other reengineering approaches described here. The notion of integrated product development really cannot be carried out and orchestrated effectively without simultaneously considering integrated process development. Thus, the concept is more commonly called *integrated product and process development* (*IPPD*).

The following definition of integrated product and process development is appropriate. *Integrated product and process development is a systems engineering and management philosophy and approach that uses functional and cross-functional work teams to produce an efficient, effective process to deploy a product or service that satisfies customer needs through concurrent application and integration of all necessary life cycle processes.* Integrated product and process development involves systems management, leadership, systems engineering processes, the products of the process, concurrent engineering, and integration of all necessary functions and processes throughout the organization to create a cost-effective product or service that provides total quality and satisfies customer needs.

Thus, IPPD is an organization's product and process development strategy. It addresses the organizational need for continual enhancement of efficiency and effectiveness in all of its processes that lead to a product or service. There are many focal points for IPPD. Twelve are particularly important.

1. A *customer satisfaction focus* is needed as a key part of competitive strategy.
2. A *focus on results* and a *product or service* are needed to bring about total customer satisfaction.
3. A *process focus* is needed because high-quality competitive products that satisfy customers and result in organizational success come from efficient and effective processes. This necessarily requires process understanding.
4. A *strategic planning and marketing focus* is needed to ensure that product and process life cycles are fully integrated throughout all organizational functions, external suppliers, and customers.
5. *A concurrent engineering focus* is needed to ensure that all functions and structures associated with fulfilling customer requirements are applied throughout the life cycle of the product to ensure correct people, correct place, correct product, and correct time deployment.
6. An *integrative engineering focus* is needed to ensure that relevant processes and the resulting processes fit together seamlessly.
7. A *teamwork and communications focus* is needed to ensure that all functional and multifunctional teams function synergistically and for the good of the customer and the organization.
8. A *people empowerment focus* is needed so that all decisions are made by qualified people at the lowest possible level consistent with authority and responsibility. Empowerment is a responsibility that entails commitment and appropriate resource allocation to support this commitment.
9. A *systems management reengineering focus* is needed for both revolutionary change and evolutionary changes in processes and product.

10. An *organizational culture and leadership focus* is needed to accommodate changed perspectives relative to customers, total quality, results and products, processes, employees, and organizational structures.
11. A *methods, tools, and techniques focus* is needed because methods, tools and techniques are needed throughout all aspects of an IPPD effort, even though they alone do not create success.
12. A *systematic measurements focus,* primarily on proactive measurements but also on interactive and reactive measurements, is needed because the organization needs to know where to go and where it is now to make progress.

All of this should bring about high quality, continual, evolutionary, and perhaps even revolutionary improvement for customer satisfaction. Each of these could be expanded into a series of questions or a checklist used to evaluate the potential effectiveness of a proposed integrated product development process and team. Although this discussion of IPPD makes it look like an approach particularly and perhaps even uniquely suitable for system acquisition, production, or procurement, it is equally applicable to the products of the RDT&E and marketing life cycles.

IPPD is a people, organizational, and technologically focused effort that is tightly linked together through a number of life cycle processes through systems management. These are major ingredients for all systems engineering and management efforts, as suggested in the information ecology (29) web of Fig. 5. The major result of IPPD is the ability to make optimum decisions with available resources and to execute them efficiently and effectively to achieve three causally linked objectives:

1. To integrate people, organizations, and technology into a set of multifunctional and networked product development teams.
2. To increase the quality and timeliness of decisions through centrally controlled, decentralized, and networked operations.
3. To satisfy customers completely through quality products and services that fulfill their expectations and meet their needs.

The bottom line is clearly customer satisfaction through quality, short product delivery time, reduced cost, improved performance, and increased capabilities. Equally supported by IPPD are organizational objectives for enhanced profit, well-being of management, and a decisive and clear focus on risk and risk management and amelioration.

Figure 6 is a suggested sequence of steps and phases to establish an integrated product and process development endeavor. The approach is not entirely different from that suggested for successful product and process development by Bowen et al. (30).

1. Understand the core capabilities and core rigidities of the organization.
2. Develop a guiding vision in terms of the product or service, the project and process, and the organization that ensures and understands the relationships between organization, customers, and process and product.
3. Push the frontiers of the organization, process, and product or service to identify and achieve the ultimate performance capabilities for each.
4. Develop leadership and an appropriate structure to manage the resulting process and product or service engineering.
5. Develop commitment at the level of organizational management, the integrated product team (*IPT*), and the individual team members to ensure appropriate ownership of the IPPD effort.
6. Use prototypes to achieve rapid learning and early evolution and testing of the IPPD concept.
7. Ensure integration of people, organizations, and technologies to attain success of the IPPD concept.

As with other efforts, this embodies the definition, development, and deployment triage that is the simplest representation of a generic systems engineering life cycle.

IPPD is a relatively new concept used very often within the US Department of Defense (31, 32). In the latter document (32) ten key tenets of IPPD are identified.

1. *Customer focus.* The primary objective of IPPD is to satisfy the needs of the customer more efficiently and effectively. Customer needs are the major determining influence on the product or service definition and the associated product lines.
2. *Concurrent development of products and processes.* It is necessary to develop processes concurrently with the products or services that they support.
3. *Early and continuous life cycle planning.* Planning for both the product or service and process begins early and extends throughout the IPPD life cycle.
4. *Maximize flexibility for optimization and use of contractor-unique approaches.* Requests for proposals should provide flexibility for optimizing and using contractor-unique processes and commercial specifications, standards, and practices.
5. *Encourage robust design and improved process capability.* Advanced robust design and manufacturing techniques that promote total quality and continuous process improvement should be emphasized.
6. *Event-driven scheduling.* The scheduling framework should relate program events to their desired accomplishments and should reduce risk by ensuring product and process maturity before actual development is undertaken.
7. *Multidisciplinary teamwork.* Multidisciplinary teamwork is essential to the integrated and concurrent development of product and process.
8. *Empowerment.* Decisions should be taken at the lowest level commensurate with appropriate risk management, and resources should be allocated at levels

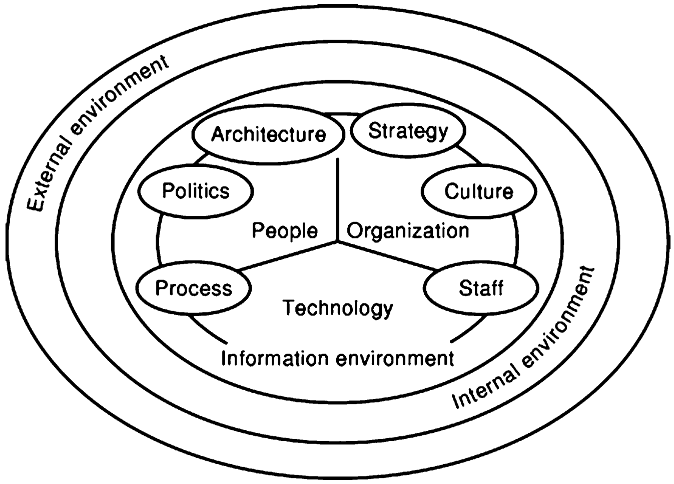

Figure 5. Information ecology web.

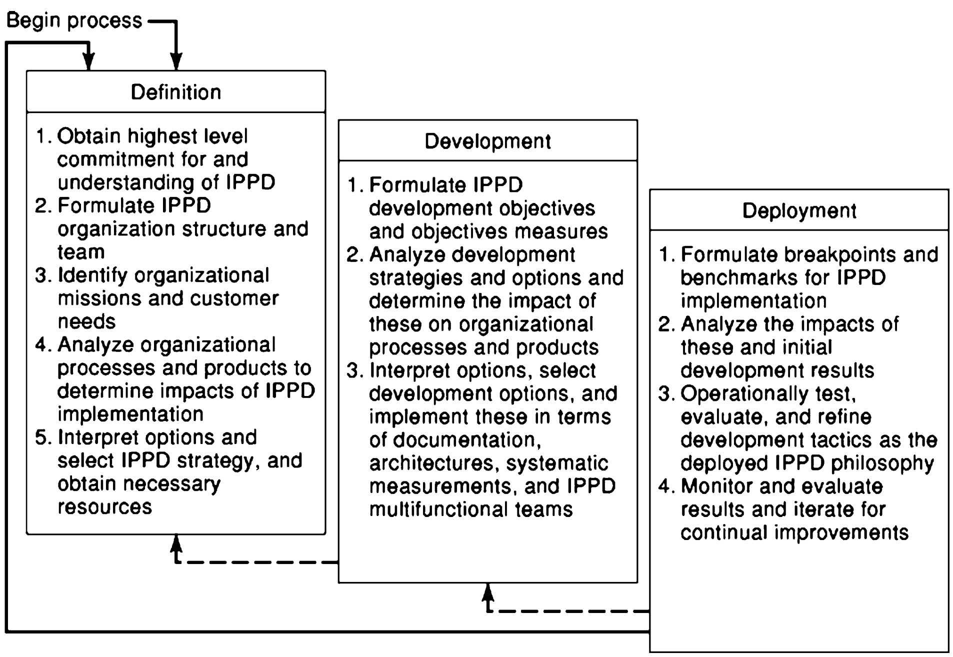

Figure 6. A simplified process to implement IPPD.

consistent with authority, responsibility, and ability. Teams should be given authority, responsibility, and resources. They should accept responsibility, manage risk appropriately, and be held accountable for the results.

9. *Seamless management tools.* A single management system should be established to relate requirements, planning, resource allocation, execution, and program tracking over the entire life cycle.
10. *Proactive identification and management of risk.* Critical cost, schedule, and technical specifications should be identified from user requirements. Systems management of risk, using appropriate metrics, should be established to provide continuing verification of achievements relative to appropriate product and process standards.

The objectives in this are to reduce time to deliver operationally functional products and services, to reduce the costs and risks of deploying systems, and to improve their quality.

Redevelopment of processes only, without attention to reengineering at a level higher than processes, may represent an incomplete and not fully satisfactory way to improve organizational capabilities if they are otherwise deficient. Thus, the processes considered as candidates for reengineering should be high-level managerial ones as well as operational processes.

REENGINEERING AT THE LEVEL OF SYSTEMS MANAGEMENT

Reengineering at the level of systems management is directed at potential change in all business or organizational processes and thereby also the various organizational life cycle processes. Many authors have discussed reengineering the corporation. The earliest use of the term *business reengineering* by Hammer (33), more fully documented in a more recent work on *Reengineering the Corporation* (34).

Hammer's definition of reengineering, *Reengineering is the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality, service and speed,* is a definition of what we will call reengineering at the level of systems management. There are four major terms in this definition:

- *Fundamental* refers to a large-scale and broad examination of virtually everything about an organization and how it operates. The purpose is to identify weaknesses that need diagnosis and correction.
- *Radical redesign* suggests disregarding existing organizational processes and structures and inventing totally new ways of accomplishing work.
- *Dramatic improvements* suggests that, in Hammer's view, reengineering is not about making marginal and incremental improvements in the status quo. It is about making quantum leaps in organizational performance.
- *Processes* represent the collection of activities used to take input materials, including intellectual inputs, and transform them into outputs and services of value to the customer.

Hammer suggests that reengineering and revolution are almost synonymous. He identifies three types of firms that attempt reengineering: those in trouble, those that see trouble coming, and those that are ambitious and seek to avoid impending troubles. Clearly, it is better to be proactive and be in this last category, rather than to be reactive and seek to emerge from a crisis.

He indicates that one major catalyst for reengineering is the creative use of information technology. Reengineering is not just automation, however. It is an ambitious and rule-breaking study of everything about the organization to effect designing, more effective and efficient organizational processes. We share this view of reengineering at the level of systems management. Our definition is similar: *Systems management reengineering is the examination, study, capture, and modification of the internal mechanisms or capability of existing system management processes and practices in an organization to reconstitute them in a new form with new features, often to take advantage of newly emerged organizational competitive requirements but without changing the inherent purpose of the organization itself.*

Figure 7 represents this concept. Life cycle process reengineering occurs as a natural by-product of reengineering at the level of systems management. This may or may not result in reengineering existing products. Generally it does. New products and new competitive strategies are each major underlying objectives of reengineering at the level of systems management, or organizational reengineering as it is more commonly called.

The work by Hammer and Champy (34) defines the forces of the three Cs,

- *Customers,* who demand customized products and services that are of high quality and trustworthy.
- *Competition,* which has intensified on a global scale in almost all areas.
- *Change,* which now becomes continuous.

These combine to require massive, discrete-level transformations in the way organizations do business. Radical and dramatic reengineering of fundamental organizational strategy and of all organizational processes is suggested as the only path to change for many organizations.

The authors are much concerned with organizational processes that have several common characteristics. Our interpretation of these is as follows:

1. The steps and phases in the process are sequenced logically in terms of earlier phase results needed for later activities. The phases are not necessarily linear. They are sequenced concurrently whenever possible to obtaining results in minimum time.
2. The various business processes are integrated throughout the organization, and often a number of formerly distinct efforts are combined to produce savings in costs and increase effectiveness.
3. Multiple versions of many processes make mass customization possible.
4. Work is shifted across organizational boundaries to include potential outsourcing and is performed in the most appropriate setting.
5. Decision making efforts become part of the normal work environment, and work is compressed both horizontally and vertically.
6. Reactive checks, controls, and measurements are reduced in frequency and importance in favor of greater use of interactive and proactive approaches.
7. There is always a point of contact, or case manager, empowered to provide service to each individual customer, and a customer need never go beyond this point of contact.
8. Organizational operations are a hybrid of centralized and decentralized structures best suited to the particular task at hand.

It is claimed that several benefits result from this. Work units change from functional departments to multifunctional, process-oriented teams. Now performers of simple tasks accomplish multidimensional work. People become empowered rather than controlled. The major needed job preparation changes, and it becomes education rather than training. The focus of measures and performance shifts to results rather than activities. Promotion or transfer to a new organizational assignment is based on ability for the new assignment and not performance in a former assignment. Values change from reactive and protective to proactive and productive. Managers become coaches as well as supervisors, executives become leaders and not just scorekeepers, and organizational structures shift away from the hierarchical to the flat. Information technology is represented as a major enabler of all of this.

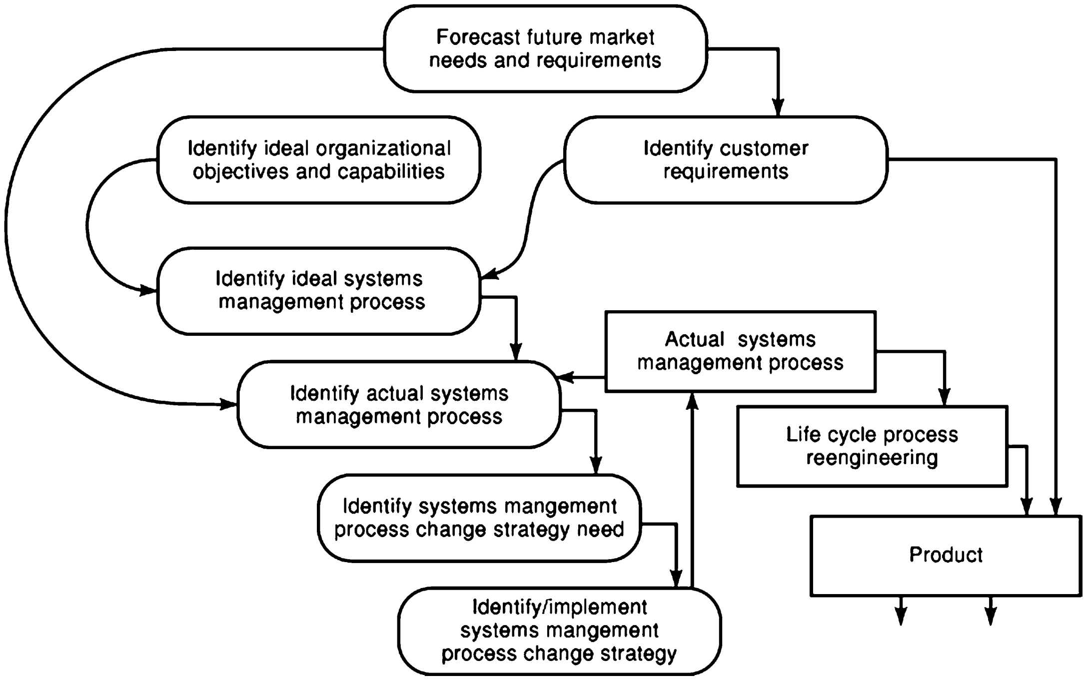

Figure 7. Conceptual illustration of reengineering at the level of systems management.

In *Reengineering the Corporation,* Hammer and Champy (34) describe a revolution in the way that organizations in the US and other developed nations generally accomplish work. This first book defined reengineering and its process components. It also suggested how jobs differ in the reengineered organization. Hammer and Stanton (35) have written *The Reengineering Revolution: A Handbook,* Champy (36) has *authored Reengineering Management: The Mandate for New Leadership,* and Hammer (37) has authored *Beyond Reengineering: How the Process Centered Organization Is Changing Our Work and Our Lives.* Each of these works extends the original efforts of Hammer and his colleagues.

All three works address the potential difficulties in implementing organizational change. Each acknowledges reengineering failures and presents strategies to overcome them. The experience of many suggest that a radical reengineering effort, or process innovation, is not always successful. The major difficulty is failure to cope with the impact of reengineering on people and their potential resistance to change. Other potential difficulties are inadequate team building and the failure of senior management to appropriately convey the need for change and to be fully aware of the human element. Hammer and Stanton's book (35) is a handbook of techniques and practical advice, and Champy (36) focuses on management as the single critical influence of reengineering success. His focus is on reengineering processes through innovation and on reengineering at the level of systems management.

The major thrusts of the Hammer and Stanton work (35) is that only senior-level managers have the breadth of perspective, knowledge, and authority required to oversee the effort from beginning to end and to overcome the resistance that occurs along the way. Senior managers must make decisions to reengineer and then create a supportive environment that results in transforming organizational culture. The reengineering team is also a major ingredient in success or failure. This team accomplishes the following:

- develops an understanding of the existing process and customer requirements to provide a *definition* of the reengineering requirements.
- identifies new process architectures and undertakes *development* of the new processes.
- provides for *deployment* of the process and the new way of doing work.

The environment of reengineering is one of uncertainty, experimentation, and pressure, and, based on these characteristics, the essential characteristics for the success of a reengineering team are identified. These include a process orientation, creativity, enthusiasm, persistence, communication skills, diplomacy, holistic perspective, and teamwork.

Dealing with the human element in an organization disoriented by the immense changes brought about by reengineering is important, and several strategies are suggested. Resistance to change is acknowledged as natural and inevitable. The imposition of a new process on people who have become attached to a familiar process creates natural resistance unless a five-step process for implanting new values is adopted.

1. Articulate and communicate the new values effectively.
2. Demonstrate commitment of the organizational leadership to the new values.
3. Hold to these values consistently.

4. Ensure that the desired values are designed into the process.
5. Measure and reward the values that the organization wants to install.

Thus, the advocacy here is centered on customers and on the end-to-end processes that create value for them. By adhering to these principles, the organization should operate with high quality, tremendous flexibility, low cost, and exceptional speed.

Champy (36) also examines the successes and failures of contemporary process reengineering innovations. He suggests that the failure of management to change appropriately is the greatest threat to successful reengineering efforts and that managers must change the way they work if they hope to realize the full benefits of reengineering. In other words, reengineering of the lower-level work details is the focus of many contemporary efforts. However, reengineering of management itself is at least as significant, and this has not yet been explored sufficiently. Such exploration and subsequent action are the major objectives in reengineering at the level of systems management. The intent here, as *In Reengineering Management,* is to identify concepts and methods that organizational administrators, managers, and leaders may use to reengineer their own executive functions for enhanced efficiency and effectiveness.

Champy begins with the impact of reengineering on managers and suggests that the greatest fear of executives is loss of control. The role of executives in a knowledge-based society is not to command or manipulate but to share information, educate, and empower. They must have faith in human beings and their ability, if led properly, to do a better job for the customer. This is called *existential authority.* To bring this about requires a change in purpose, culture, processes, and attitudes toward people. Champy suggests that managers must focus on the answers to four questions to enable these changes.

1. What is the purpose of the organization?
2. What kind of organizational culture is desired?
3. How does the organization go about its work?
4. What are the appropriate kinds of people for the organization?

He suggests that management processes provide support for management reengineering and defines new core management processes for the reengineered executive.

As a consequence of reengineering at the level of systems management, hierarchies are flattened. Culture rather than structure is more of a determinant of the way the organization runs. A major need is for managers to organize high-performance, cross-functional teams around the needs of changing product lines or processes. Profit and principle must be congruent. Five core management processes are identified. Each of them potentially needs to be reengineered to harmonize with the core capabilities and mission of the organization.

1. *Mobilizing* is the process through which an organization, including its human element, is led to accept the changes brought about by reengineering.
2. *Enabling,* or empowering, involves redesigning work so that humans can use their capabilities as much as possible and must foster a culture that motivates people to behave the way the organization needs them to behave.
3. *Defining* is the process of leadership through continual experimentation and empirical efforts. This includes the development of experiential learning from these efforts and learning to act on what is learned from them.
4. *Measuring* is focused on identifying important process results, or metrics, that accurately evaluate organizational performance.
5. *Communicating* involves continually making the case for changes that lead to organizational improvement and being concerned with the "what" and "how" and also with the impacts of actions on employee lives. As suggested by many, managers are now coaches and must provide tools needed to accomplish tasks, remove obstacles hindering team performance, and challenge imaginations by sharing of information. This relates strongly to empowerment and to trust building, which is a goal of communications.

This "people-focused management" requires "deep generalists," who respond to changing work demands, changing market opportunities, evolving and changing products and services, and changing demands of customers. Of course, it is necessary that these generalists also bring deep expertise in a specialty area and well-established skills to the organization.

PERSPECTIVES ON REENGINEERING

This discussion of reengineering suggests that reengineering can be considered at three levels: systems management, life cycle processes, and product. These three levels, when associated with appropriate methods, tools, and metrics, constitute a relatively complete conceptual picture of systems engineering efforts, as shown in Fig. 8. The major purpose of reengineering at any level is to enable an organization to produce a better product at the same or lower cost that performs comparably to the initial product. Thus, reengineering improves the competitiveness of the organization in coping with changing external situations and environments. An organization may approach reengineering at any or all of these levels from any of three perspectives:

- *reactive* because the organization realizes that it is in trouble and perhaps in a crisis, and reengineering is one way to bring about needed change;
- *interactive* because it wishes to stay abreast of current changes as they evolve; or
- *proactive* because it wishes to position itself now for changes that it believes will occur in the future and to emerge in the changed situation as a market leader.

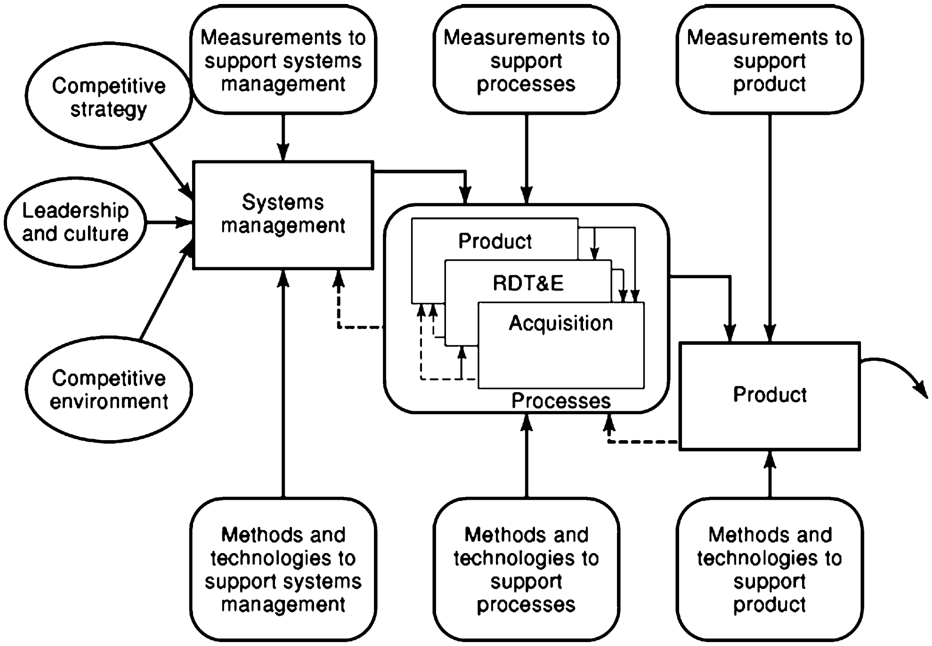

Figure 8. Conceptual model of systems engineering morphology.

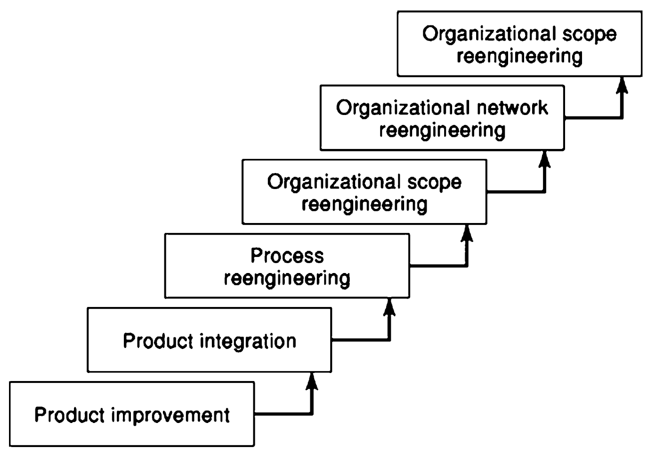

Figure 9. Representation of improvements at the level of products, processes, and systems management through reengineering.

Reengineering could be approached from an inactive perspective, although this suggests not considering it at all, and this is likely to lead to failure to adapt to changed conditions and requirements.

Reengineering at any level—product, process, or systems management— is related to reengineering at the other two levels. Reengineering can be viewed from the perspective of the organization fielding a product as well as from the perspective of the customer, individual or organizational, receiving the product. From the perspective of either of these, it may well turn out that reengineering at the level of product only may not be fully meaningful if it is not also associated with, and generally driven by, reengineering at the levels of process and systems management. For an organization to reengineer a product when it is in need of reengineering at the systems management or process levels is almost a guarantee that a reengineered product will not be fully trustworthy and cost-efficient. An organization that contracts for product reengineering when it is in need of reengineering at the levels of systems management and/or process is asking for a technological fix and a symptomatic cure for difficulties that are institutionally and value related. Such solutions are not really solutions at all.

There are potential needs for integrated reengineering at the levels of product, process, and systems management. Product reengineering may consume significant resources; the combined resources needed for systems management and process reengineering can also be substantial. Resources expended on product reengineering only, and with no investigation of needs at the systems management and process levels, may not be wise expenditures from the perspective of the organization producing the product or the one consuming it.

In an insightful article, Venkatraman (38) identifies five levels for organizational transformation through information technology. We can expand on this slightly through adoption of the three levels for reengineering we have described here and obtain the representation shown in Fig. 9. This figure shows our representation of these five levels: two for organizational reengineering, two for product reengineering, and one for process reengineering. Organizational reengineering is generally revolutionary and radical, whereas product reengineering is usually evolutionary and incremental. Process reengineering may be at either of these extremes.

Venkatraman notes technological and organizational enablers and inhibitors that affect desired transformations at both evolutionary and revolutionary levels of transformation. Technological enablers include increasingly favorable trends in cost effectiveness for various information technologies and possibilities of enhanced connectivity. Technological inhibitors include the lack of currently established, universally accepted standards and the rapid obsolescence of current technologies. Organizational enablers include managerial awareness of the need for change in existing leadership. Organizational inhibitors include financial limitations and managerial resistance to change. Although both product reengineering and organizational reengineering ultimately lead to change in organizational processes, changes for the purpose of producing a product with greater cost-effectiveness, quality, and customer satisfaction generally differ from and are more limited than those for improving internal responsiveness to the satisfaction of present and future customer expectations. Organizational network and organizational scope are at the highest level here, because efforts at these levels are of much concern relative to information technology and associated knowledge management efforts today, including contemporary efforts involving systems integration and architecting (39).

Effective management pays particular attention to technology and to the human elements in the organizations and the environments in which they are embedded. In a recent work (40), eight practices of exceptional companies are described:

- balanced value fixation
- commitment to a core strategy
- culture-system linkage
- massive two-way communication
- partnering with stakeholders
- functional collaboration
- innovation and risk
- never being satisfied

and guidelines are presented that enable the enduring human asset management practices that make such efforts as reengineering long-term successes. The *human element* is a major part of reengineering because the purpose of technology is to support human endeavors.

Reengineering efforts continue to this day, although the name is sometimes changed to reflect contemporary issues better. One current phrase, which is in many ways reengineering of the term reengineering, is that of enterprise transformation (41). This does not, in any sense suggest that such terms are merely contemporary buzzwords as they reflect a needed critical awareness that the vast majority of change, if it is to be meaningful, must truly start from and consider issues at the highest, or enterprise level.

BIBLIOGRAPHY

1. A. P. Sage *Systems Management for Information Technology and Software Engineering*, New York: Wiley, 1995.
2. A. P. Sage Systems reengineering, in A. P. Sage and W. B. Rouse (eds.), *Handbook of Systems Engineering and Management*, New York: Wiley, 1999.(Second edition in Press 2007).
3. M. G. Rekoff, Jr. On reverse engineering, *IEEE Trans. Syst. Man Cybern.*, **SMC-15**: 244–252, 1985.
4. Software Engineering Glossary, *IEEE Software Engineering Standards*, New York: IEEE Press, 1991.
5. IEEE Standard for Software Maintenance, P1219/D14, New York: IEEE Standards Department, 1992.
6. E. Chikofsky J. H. Cross Reverse engineering and design recovery, A Taxonomy, *IEEE Software*, **7** (1): 13–17, 1990.
7. J. H. Cross II E. J. Chikofsky C. H. May, Jr. Reverse engineering, in M. C. Yovitz (ed.), *Advances in Computers*, Vol. 35, San Diego, CA: Academic Press, 1992, pp. 199–254.
8. R. S. Arnold (ed.) *Software Reengineering*, Los Altos, CA: IEEE Computer Society Press, 1993.
9. H. M. Sneed Planning the reengineering of legacy systems, *IEEE Software*, **12** (1): 24–34, 1995.
10. A. P. Sage *Systems Engineering*, New York: Wiley, 1992.
11. H. M. Sneed Planning the reengineering of legacy systems, *IEEE Software*, **12** (1): 24–34, 1995.
12. W. M. Ulrich Re-engineering: Defining an integrated migration framework, in R. S. Arnold (ed.), *Software Reengineering*, Los Altos, CA: IEEE Computer Society Press, 1993, pp. 108–118.
13. M. R. Olsem Preparing to reengineer, *IEEE Comput. Soc. Reverse Eng. Newsl.*, December 1993, pp. 1–3.
14. H. A. Muller Understanding software systems using reverse engineering technologies research and practice, Proc. Int. Conf. Software Eng., Berlin, Germany, March 1996.
15. P. Samuelson Reverse engineering someone else's software: Is it legal?, *IEEE Software*, **7** (1): 90–96, 1990.
16. V. Sibor Interpreting reverse engineering law, *IEEE Software*, **7** (4): 4–10, 1990.
17. E. M. Hall *Managing Risk: Methods for Software Systems Development*, Reading, MA: Addison-Wesley, 1998.
18. J. D. Ahrens N. S. Prywes Transition to a legacy and reuse based software life cycle, *IEEE Comput.*, **28** (10): 27–36, 1995.
19. R. Wheeler R. W. Burnett A. Rosenblatt Concurrent engineering, *IEEE Spectrum*, **28** (7): 32–37, 1991.
20. A. Kusiak (ed.) *Concurrent Engineering: Automation, Tools, and Techniques*, New York: Wiley, 1993.
21. B. Prasad *Concurrent Engineering Fundamentals: Integrated Product and Process Organization*, Englewood Cliffs, NJ: Prentice-Hall, 1996.
22. B. Prasad *Concurrent Engineering Fundamentals: Integrated Product Development*, Englewood Cliffs, NJ: Prentice-Hall, 1996.
23. C. C. Andrews N. S. Leventhal *FUSION-Integrating IE, CASE, and JAD: A Handbook for Reengineering the Systems Organization*, Englewood Cliffs, NJ: Prentice-Hall, 1993.
24. K. Kronlof (ed.) *Method Integration: Concepts and Case Studies*, Chichester, UK: Wiley, 1993.
25. D. Schefstrom G. van den Broek (eds.) *Tool Integration: Environments and Frameworks*, Chichester, UK: Wiley, 1993.
26. A. P. Sage J. D. Palmer *Software Systems Engineering*, New York: Wiley, 1990.
27. D. E. Carter B. S. Baker *Concurrent Engineering: The Product Development Environment for the 1990s*, Reading, MA: Addison-Wesley, 1992.
28. J. Fiksel Computer-aided requirements management for environmental excellence, Proc. Natl. Council Syst. Eng. Annu. Meet., Alexandria, VA, July 1993, pp. 251–258.
29. T. H. Davenport *Information Ecology: Mastering the Information and Knowledge Environment*, New York: Oxford Univ. Press, 1997.
30. H. K. Bowen et al. (eds.) *The Perpetual Enterprise Machine: Seven Keys to Corporate Renewal Through Successful Product and Process Development*, New York: Oxford Univ. Press, 1994.

31. Air Force Material Command Guide to Integrated Product Development, May 25, 1993.
32. DoD Guide to Integrated Product and Process Development, Office of the Undersecretary of Defense for Acquisition and Technology, February 1996.
33. M. Hammer Reengineering work: Don't automate, obliterate, *Harvard Bus. Rev.*, **68** (4): 104–112, 1990.
34. M. Hammer J. Champy *Reengineering the Corporation: A Manifesto for Business Revolution*, New York: Harper Business, 2004.
35. M. Hammer S. Stanton *The Reengineering Revolution*, New York: Harper Business, 1995.
36. J. Champy *Reengineering Management: The Mandate for New Leadership*, New York: Harper Collins, 1995.
37. M. Hammer *Beyond Reengineering: How the Process-Centered Organization is Changing Our Work and Our Lives*, New York: Harper Business, 1996.
38. N. Venkatraman IT-enabled business transformation: From automation to business scope redefinition, *Sloan Manage. Rev.*, **35** (2): 73–88, 1994.
39. A. P. Sage C. L. Lynch Systems integration and architecting: An overview of principles, practices, and perspectives, *Syst. Eng.*, **1** (3): 1998.
40. J. Fitz-Enz *The 8 Practices of Exceptional Companies: How Great Organizations Make the Most of Their Human Assets*, New York: AMACOM, 2005.
41. W. B. Rouse (Ed.), *Enterprise Transformation: Understanding and Enabling Fundamental Change*, John Wiley and Sons, Hoboken NJ, 2006.

ANDREW P. SAGE
George Mason University,
Fairfax, VA

Table 1. Attributes to Consider in Analyzing Reengineering Project

Current annual maintenance cost
Current annual operating cost
Current annual business value
Predicted annual maintenance cost after reengineering
Predicted annual operating cost after reengineering
Predicted annual business value after reengineering
Estimated reengineering costs
Estimated reengineering calendar time
Reengineering risk factors
Predicted annual maintenance cost after redevelopment
Predicted annual operating cost after redevelopment
Predicted annual business value of the new system
Estimated redevelopment costs
Estimated redevelopment calendar time
Redevelopment risk factors
Expected life of the reengineering system