

ERP/SCM/eCommerce: Package Implementation Methods and Considerations

A White Paper

by **Richard Luetngen**
ThoughtWing

This paper was developed to provide general background to assist clients in decisions related to implementing enterprise-wide ERP (Enterprise Resource Planning) software package suites, although the concepts and techniques apply as well to SCM (Supply Chain Management), CRM (Customer Relationship Management) and other complex eCommerce packaged software suites. *ERP* evolved as an enterprise-wide view of MRP, and primarily automates the transaction processing requirements of financial, distribution and manufacturing business processes.

Please note that this paper presents professional opinions intended to apply generally and that clients must take appropriate care to evaluate them in light of their specific needs. ThoughtWing makes no representations, warranties or guarantees of any sort as to the applicability of the opinions presented in this paper to the specific needs of any client.

The paper is organized as follows:

1. Overview
2. Staging and Change Management
 - 2.1 Staging and Change Management
 - 2.2 Preliminary Business Process Reengineering Work
 - 2.3 Prototyping and Testing
 - 2.4 Roll-Out
 - 2.5 Post-Implementation Review.

1. Overview

Implementing complex software package suites can be the most complex and riskiest endeavor undertaken by a company. This is due primarily to the fact that the effort touches, perhaps drastically affects, almost every business process within the enterprise. Change on such a scale can intimidate personnel, who fear for their places within the organization, and who may sabotage efforts; and, even in highly cooperative environments, such fundamental change is difficult to accomplish.

The remainder of this *Overview* discusses major categories of risk attendant to complex package implementation, technical and political, as well as techniques and actions available to a management team to minimize and manage the risks. Following the *Overview* is a set of phases required to actually introduce fully functioning suites into an organization. In each phase, major activities, deliverables and required actions are discussed, as well as the risks and some available remedial options relevant to the phase.

The major categories of risk discussed below are:

- Complexity
- Change
- Software Modification
- Participation of Your People
- Effective Sponsorship/Organizational Stability
- Legacy Systems Enhancement.

1.1 Complexity

Complexity can be generally defined as the interaction of many moving, interrelated parts. This definition describes complex software package implementation well. Many things need to happen in such an effort, often concurrently, including:

- Reengineered business processes (if necessary)
- New policies & procedures (if necessary)
- New computers & network environment (if necessary)
- Parameterized software product modules
- Converted historical data
- Modified product software (if necessary)
- Custom-developed software (if necessary)
- Interfaced legacy applications (if necessary)
- Trained user and technical communities.



Effectively orchestrating all sets of moving parts is a difficult exercise, requiring extensive complex program management experience. Indeed, experience on the part of program management is probably the single most important means available for effectively managing risk of failure attendant to complexity.

Next, team composition and team structure below program management are extremely important. Are roles clear? Are skills complementary? Is each business process “owned” by a senior internal individual thoroughly familiar with his or her business process and authorized to approve change?

Finally, the program plan is crucial. Complexity is best managed by organizing it effectively, and a strong plan based on a proven methodology for carrying out key activities and producing key deliverables is central to minimizing the risk of missing important steps or performing them badly.

1.2 Change

The degree of change required of an organization to implement and effectively employ complex package suites varies widely but is usually significant, sometimes very extensive. All types of change are involved, including technological change, organizational change and business process change.

Change is a major element of risk because people fear it and *will* resist it. Regardless of the commitment of management to effect change, people who perceive their interests threatened by it will support its failure.

The environment where the greatest degree of required change is often observed is the multi-divisional enterprise where the software suite is intended to be a means of standardizing business processes across divisions: in such environments, everyone must change almost everything at least to some degree.

So, how is this risk managed? By recognizing it, analyzing it and preparing for it. These activities involve a discipline known as *Change Management*. This discipline seeks first to identify the barriers to change within an organization, then to prepare for the change by developing specific programs that address and lower each barrier.

Barriers to change can include a workforce with obsolete skills, which will produce much fear on their part regarding their role in a new environment. Key people will fear loss of their value to the organization, with predictable and dire career consequences. They will resist the change, sometimes in very subtle ways difficult to detect and counter. Enough resistance and the effort will fail, very expensively. Yet a selective program of re-training and inclusion of key people in the *process* of change can often lower this barrier substantially.

Other barriers to change can be quite simple, such as inability of current facilities to support new technology: insufficient physical space, inadequate internal wiring, etc. Such barriers need to be anticipated too, but effective planning typically is sufficient to manage risk in such cases.

One of the central elements in any change management exercise is a *Communication Plan*. Such a plan identifies all the stakeholders in an organization, those individuals whose interests are affected by the new environment in some way; and, the degree to which their interests are affected. Often, this can include almost all personnel. Then, means of communicating status of the program are developed and put in place. Such means can include a newsletter, regular emails, an internal web page or regularly staged meetings with particularly important stakeholders. By whatever means, keeping all stakeholders up to date on progress and results is key to avoiding unnecessary delay and potential resistance.

In a small package software implementation the extent of change management might be the part-time efforts of one knowledgeable professional with assistance from support personnel such as an administrative assistant. In a very large implementation, the effort could be a discrete team managed full-time by a very senior change management professional.

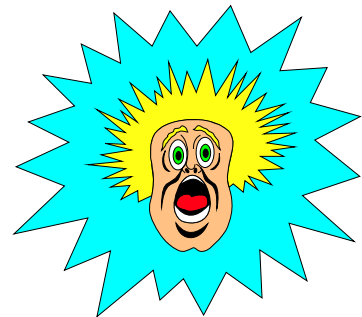
1.3 Software Modification

The temptation to modify packaged software to meet the demands of a particular business process almost always is intense.

This occurs because, while often very flexible, the software may not accommodate every variation in process across an enterprise, particularly multi-divisional enterprises; and, important people will champion the need to retain each variation.

But modification will:

- complicate an already very complex process and increase the risk of failure
- complicate upgrades in software version, thereby driving up, perhaps dramatically, the cost of ownership and use of the software
- require you to maintain the modification yourself at your expense rather than require the vendor to add functionality and flexibility to a standard product at his expense.



I can't change the process ...
Change the **software!!!**

This problem may be addressed by scrutinizing the business process or variation in question and determining:

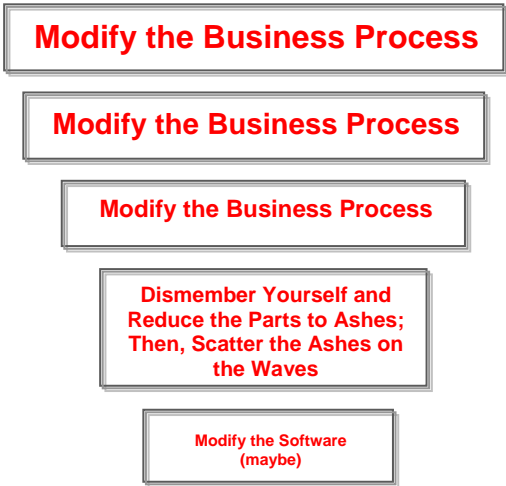
- Does its uniqueness truly support competitive differentiation? Does it honestly lower cost or improve quality relative to more easily supported alternatives? Does it support a level of customer service not obtainable by competitors employing different processes?
Or is its apparent importance really tactical convenience for a number of people who find change inconvenient?
- Does it represent a best-in-class process alternative demonstrated in other companies to yield optimal results?

Often, this structured approach can get beyond the emotion of the moment to the real issues. People honestly desirous of supporting legitimate management commitment to beneficial change will support the decision when given compelling reasons.

It helps when management anticipates this problem and states a policy, including it as part of the Communication Plan, as one of several *guiding principles* for the implementation. The policy might be expressed as the example, to the right, with the prescription:

Customization is very expensive and often not necessary, yet it's always tempting. Accordingly, please observe the following principles when considering it, in descending order of priority:

All this said, customization sometimes is unavoidable. However, without a highly structured process in place to filter unnecessary customization, the first one often has the affect of opening the floodgates. This should be anticipated and its dangers discussed openly with all stakeholders.



1.4 Participation of Your People

Key people will need to participate fully in the process, which can disrupt business. This is often the most poorly appreciated risk by managements about to embark on major implementations, and it can be catastrophic in its effects.

Expect that no less than ten percent of a key person's time will need to be dedicated at *any* time to the implementation. Up to eighty percent of that key person's time may need to be dedicated during particularly intensive portions of the implementation, such as process reengineering and testing.

When fully understood, these demands will cause alarm:

- I have a business/department to run! I'd like to participate, but... how can I do both?!
- Am I automating myself out of a job? Where will I fit in the new picture?

Often, a key person spends as little as twenty percent of her time performing the *high-value-added* activities that make her key – maintaining a crucial customer relationship, solving rare but critical problems, coming up with creative new ideas, etc. As much as 80% of her time can be spent in non-key or *commodity* tasks, such as approving others' actions or reviewing or preparing reports. The commodity tasks may have evolved over time to justify a job title or as a means to fill a day until demand surfaced to employ high-value-added skills.

Shift the commodity tasks to someone else or recognize their nature and do away with them entirely. This could free up substantial time that can be dedicated to the implementation. Not performing such tasks does not detract from the person's key status – the relationships still need to be maintained, the problems solved, the ideas generated; this is where the person's value lies.

Key people also need to be shown where they fit in the new picture in order to motivate them to full and effective participation. This means that management must have a vision for the new environment and understand where people fit in the vision.

1.5 Effective Sponsorship/Organizational Stability

Success can be threatened by ineffective sponsorship of a complex package software implementation within a company.

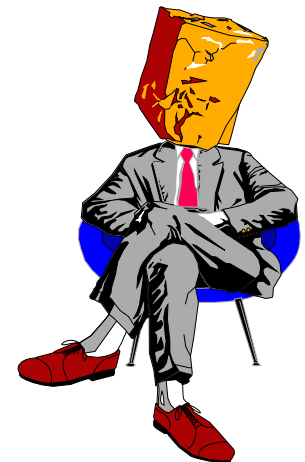
If the *sponsor* is not a major influencer within the company, too much time may be spent in achieving closure on contentious issues and in acting. If the sponsor is positioned too low in the food chain, decisions, once taken and acted upon, may be revisited later as pressure increases to protect individual agendas.

Such conditions could have dire consequences:

- They could increase cost, perhaps disastrously
- They will consume political capital required for other things
- They will destroy morale.

A strong sponsor will help match accomplishments to expectations.

Not The Best Choice



“This is way too political ... anything I say is off the record.”

He or she should:

- be a senior executive of the company
- be directly dependent on the success of the implementation and committed to it
- have substantial authority over large numbers of people
- be stable – not about to be lateral-arabesqued into “special projects”
- be a strong doer
- *not* be the person in charge of the IT/MIS function – that individual already has enough riding on success or failure and will be perceived as protective of IT interests which may not necessarily be perceived as business interests.

Stability of the program team is just as essential if not more so. Without such consultant/integrator stability, the program stands a very good chance of failing, at a very high cost. Client management should deal with constant changes in the integrator(s) program management team very quickly and very sternly. With the first significant change, the consultant/integrator should be put on notice that his continued involvement is under review; and with the next it could be terminated. This may seem draconian (and expensive), but the war stories of failed implementations due to revolving door consulting management are legion. A re-staged implementation is less expensive than a failed one, but it requires strong client leadership.

1.6 Legacy Systems Enhancement

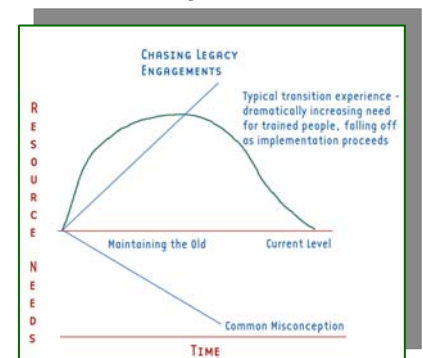
Often, a poor balance is struck between implementing the new and maintaining the old, yet an optimal balance is very important. Other than an optimal balance will drain scarce resources into unproductive paths and provide distractions from the important objectives.

The graph to the right illustrates some of the temptations inherent in this balancing act.

A common misconception is that all enhancements and almost all maintenance activity to the legacy, or current systems and other technology infrastructure, will be dramatically reduced while the new environment is put in place. Yet, the world does not stop revolving while new systems are implemented.

New products are developed, competitors come up with new ideas that need to be countered, new channels of distribution are identified, and customers come up with new service demands, and on and on. You often need to address such things timely, not wait “until the new systems are in”. This may require enhancement to current systems, even while you are implementing a new environment.

Yet, you could also yield to the costly temptation to chase such enhancements. How do you differentiate between real and unnecessary in a high-pressure environment?



As with other risks, you anticipate the issue and plan for it:

- First, establish a policy that only clearly beneficial enhancements of any size to legacy components will be considered;
- Then, establish a committee to review requested legacy enhancements against clearly stated criteria of what is beneficial.

In the late seventies and early eighties, Lee Iacocca, struggling with the survival of Chrysler, established that no enhancements would be considered unless they could be demonstrated to have a six-month payback. Result: No enhancements. Another, broader criterion could be clearly demonstrable need to counter competitive threat where failure to do so would result in loss of market share.

Another problem that crops up is the desire on the part of key IT professionals to be involved in the new implementation (not surprising, since the new systems represent future jobs while the old systems will go away); yet, often they are needed to simply maintain the legacy environment. What to do?

One option is to put in place specialized outsourcing teams, *legacy* systems outsourcing teams, typically of consultants who go away when the old systems go away. While not cheap, this solution has the double advantage of permitting re-training of internal personnel in the new systems while keeping them motivated and available to address serious legacy problems. And at least part of the cost is rationalized by reduced need for consultants on the implementation team.

The moral inherent in this risk category is *balance*, in the use of scarce and expensive resources as well as in their mobilization and focus.

The remainder of this white paper deals with the major phases of an ERP implementation, as well as with major deliverables and specific risks that must be managed in each phase. The categories discussed are:

- 2.0 Complex Package Implementation Phase Overview
- 2.1 Staging and Change Management
- 2.2 Preliminary Business Process Reengineering Work
- 2.3 Prototyping and Testing
- 2.4 Roll-out
- 2.5 Post-Implementation Review.

2.0 Complex Package Implementation Phase Overview

Implementation of complex package software cannot have a completely standard method of proceeding, since each commercially available product, particularly ERPs, has a vendor more or less insistent that their methodology for implementing the product must be used and the methodologies differ.

They can differ in major ways. SAP, for instance, insists that such an implementation must be conducted within a structured framework of active business process reengineering. Other vendors either do not deal extensively with business process or are far less insistent, dealing with process variations unsupported by their software as they encounter them.

Consultants from major organizations, such as the Big 5, often have their own methodologies as well, which can differ very dramatically from the vendor's. Often, a hybrid emerges to allow mixed teams of consultants to operate with tools at least partially familiar to all participants.

The client company's management should take charge of any such methodological approach, since they will bear the risks of implementation, pay the costs and live with the results. How do you do this?

You begin by clearly understanding your objectives. Do you wish to transform your business processes into optimal models or do you wish to simply implement new software with a minimum of fuss and process disruption? Or something in between, tinkering with some processes but desiring to leave others untouched (if possible)? The degree to which a proposed methodology or methodologies support your objectives is actually a matter more appropriate to product selection than to product implementation. Moral: understand your objectives early. State your objectives to proposing consultants and require that they describe how their approach supports them.

Since there is no completely standard approach to implementing complex suites, the remainder of this white paper deals with major classes of activities. Where appropriate, considerations relevant to different approaches are discussed and advice given as to how to evaluate options.

2.1 Staging and Change Management

The primary objectives of *Staging* and *Change Management* are to:

- Determine the order of implementation
- Identify team members and organize and train them
- Anticipate barriers to change, formulate remedial programs and produce a Communication Plan
- Produce a Program Plan.

An ERP generally consists of module groups that support financial processes, manufacturing processes and distribution processes. Determining the order of implementation of the module groups is the first step to staging an implementation. In large, complex organizations, planning the implementation order of the actual modules themselves (e.g., general ledger, accounts payable, etc. within the financials group) can be equally important.

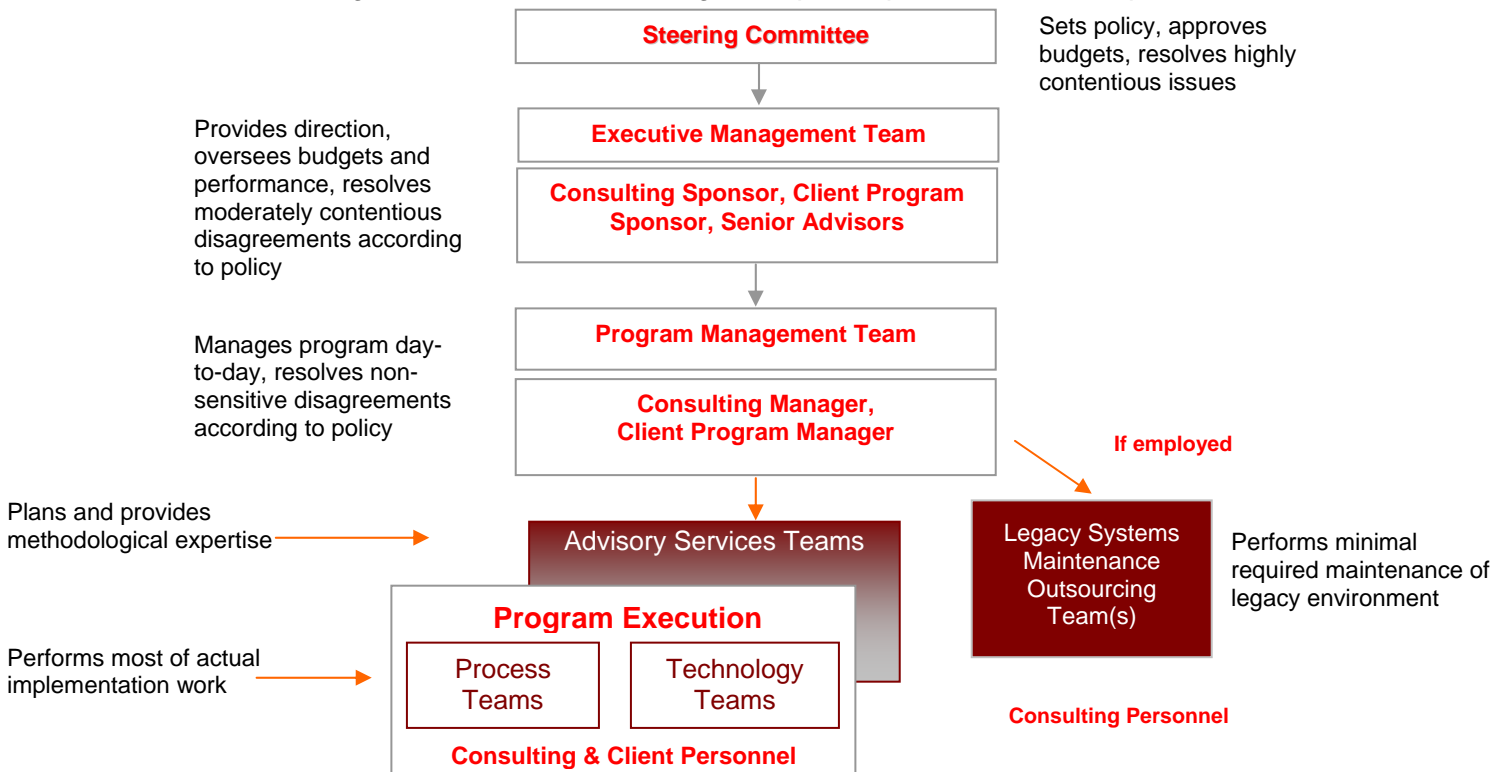
In small, relatively simple implementations with little desired business process reengineering, a “big bang” approach of implementing all modules at the same time might be attempted. But in most cases a phased implementation is determined in order to mitigate excessive risk and to spare extreme demands placed on internal personnel, many who may need to participate in the implementation of modules *across* groups.

Financials are usually chosen to be first because they tend to be the easiest to implement and the least contentious – for example, financials rarely come under customization pressure.

For similar reasons, manufacturing tends to be implemented last, since it is usually the most difficult group to implement. Implementing the easiest first permits the team structure and members to shake out and gain experience in the process before attempting the usually more difficult distribution and manufacturing groups.

However, there could exist compelling business reasons for choosing a different implementation order. In any event, in any phased implementation consideration should be given to the need to build temporary interfaces to legacy applications. While they will exist only as long as the mixed legacy-new environment exists, they should still be reliable.

Identification of team members and their organization and training comes next. If adherence to a budget and a timetable are important in the implementation (as they are to most companies), then highly matrixed organizations are not indicated: clear lines of authority are optimal. If the company culture cannot practicably support such a structure, then additional time and capital should be allocated to accommodate the needs of broad consensus on all matters. One workable organizational structure for large, complex implementations is depicted as follows:



A *Steering Committee* should be composed of executive and departmental management. They set overall policy, such as issues they believe might prove so contentious or sensitive that they wish to resolve them rather than allow resolution at a lower level.

Both the consulting sponsor and the client program sponsor should sit on the Steering Committee, assisted by the consulting manager and client program manager, providing the committee with regular updates on program status.

The *Executive Management Team* should consist of a consulting sponsor, who is responsible to the client for the quality of deliverables produced by and the professional competence of the consultants assisting the implementation, and a client program sponsor, who has overall responsibility to the Steering Committee for the success of the program.

If there are multiple consulting parties involved, each should be represented. Final decision-making authority at this level rests with the client program manager.

This team provides overall direction to the implementation, setting policy consistent with overall policy set by the Steering Committee and resolving all but the most contentious disputes. This is where the seniority of the client program sponsor is key: if that person is insufficiently influential, all disputes will become irresolvable at her level, and progress will break down at the Steering Committee level. This team also may include senior advisors, often consultants, who provide very senior expertise in specialized areas, such as best-in-class process alternatives and industry expertise.

The Management Team consists of a day-to-day consulting program manager as well as a day-to-day client program manager. Both positions are full-time in large implementations. This team acts jointly to approve tactical plans, assure that deliverables are produced timely and of appropriate quality, and resolve most issues. The client program manager is usually drawn from senior IT/MIS ranks, and actually could be the Chief Information Officer.

Advisory teams, usually consisting of consultants, provide the following services:

- Program planning and management techniques
- Methodology
- Change management
- Technology infrastructure and connectivity planning
- Workgroup facilitation
- Process change expertise and techniques
- Test planning, environment definition and execution.

Program Execution teams perform most of the actual work of the implementation, which has been planned by the Advisory teams and is managed by the Program Management Team.

The Process teams are drawn substantially from internal client personnel, organized by process specialty, while the Technology teams, dealing with product and technology infrastructure issues, are usually drawn from consulting ranks for product expertise and from internal IT/MIS ranks for technology infrastructure expertise.

A “Process Owner”, a key individual with personal authority who is thoroughly familiar with the relevant business process, captains each Process Team. This individual *must* be authorized to approve process changes that are not highly contentious. Inability to make decisions in this regard will bog down program management to an unsupportable degree.

The specific activities performed by Process teams, which drive required skills, include:

- Definition of business process, transaction processing and information requirements
- Actual prototyping of products against requirements
- Participation in business process and/or software modifications
- Training
- Participation in testing and transition to a production environment
- Roll-out of prototype(s) to the organization.

Specific activities performed by Technology teams also drive required skills, and include:

- Equipment and telecommunication platform(s) definition and sizing
- Operating system(s) definition, maintenance and tuning
- Database platform(s) definition, maintenance and tuning
- CASE (Computer Aided Software Engineering) environment and tools definition and use
- Data conversion planning and execution
- Interface planning and execution
- Software module modification
- Application report generation (unless this responsibility is assigned the Process teams due to access to a highly user-friendly report generation tool)
- Test execution
- Production environment planning and documentation
- Legacy systems maintenance (may also be supported by a separate outsourcing team).

Team sizes depend directly on the size and complexity of the implementation, and on the desired level of confidence in the implementation result, which is to say the desired degree of risk protection. A small, relatively simple implementation may require all of the services of a large one but might actually be accomplished by very few people acting in multiple capacities. A very large, complex implementation may take in a substantial percentage of management and supervisory personnel, as well as numerous consultants. Once a program plan is developed required team sizes and specific required skills could be determined by the amount and nature of actual work that needs to be performed within a specific time period.

The product vendor or reseller usually performs most product training of team personnel, and focuses on product features and implementation methodology. If significant business process reengineering is anticipated, either the vendor or another consulting partner may provide training on the means by which business processes are defined, analyzed and reengineered.

Concurrent with most of the activities related above, a (usually) small *Advisory Team* of change management professionals is interviewing company personnel and reviewing organizational structures and physical facilities, in order to anticipate barriers to change and to formulate remedial programs.

Such barriers can include major cultural attributes, such as fear of change because the organization has been stable for many years with employees possessing substantial tenure; or, as simple as the outright refusal of a key individual to participate in the change; or, again as complex as the clear and unavoidable obsolescence of skills possessed by large numbers of employees who may leave but are necessary through transition to the new environment. In some highly complex and sensitive cases, industrial psychologists are used to provide specialized insight to the team's work.

The primary objectives of the team always are to identify the substantive issues that could delay or complicate the implementation, or cause it to fail, to identify the implications of each change barrier, and to develop programs to mitigate the risks and lower the barriers; to develop contingencies in the event that the primary programs do not succeed, and to develop measurement schemes to determine if the programs are on track.

A secondary but important objective is development of a Communication Plan whose intent is to communicate implementation status to all stakeholders in the implementation. Once completed, the Program Management Team performs actual management of the plan.

The Program Plan is developed early in draft, usually by the Consulting and Client program managers, assisted by Advisory team members. This draft version informs subsequent planning activities, such as team selection, sizing, organization and training. As all planning activities are completed and approved, the Program Plan is finalized and becomes the blueprint for the implementation. It includes the following items:

- A statement of program objectives (*a mission statement*), and a definition of policy and guidelines governing the implementation
- A definition of all activities and significant tasks, significant deliverables and targeted timeframes for their delivery
- A statement of budgeted capital required and how that capital will be consumed and disbursed
- A clear statement of organization and lines of authority
- Specification of an infrastructure for managing the implementation, including the means by which decisions are made and issues resolved, and the means by which program status will be measured, reported and managed.

Highly detailed tasks and deliverables contributing to the larger tasks and deliverables typically can be defined on-going, as part of a short-interval scheduling process employed by team leaders to manage two-week time periods.

A few words are warranted regarding use of tools for project management, which capture, summarize and report task-and-deliverable-based status, as well as financial data, by assigned individual. Be careful. Such tools, while extremely useful management aids, tend to be very expensive to use during the course of an implementation program, often requiring a team to input data, prepare reports, interpret results and inform participants. Often, this burden is poorly understood at the outset of an implementation program. For smaller programs, it could make more sense to use such a tool to initially capture the scheduling elements of a Program Plan, in order to use the reports, but to determine ongoing status manually. For very large, programs in which large amounts of capital are at risk, use of such tools probably represents a small relative investment for added insurance, but the cost should be clearly understood up front.

2.2 Preliminary Business Process Reengineering Work

Whether or not it is anticipated, most complex package software implementations encounter some need to reengineer business processes, even when substantial software modification is expected. Some implementations actually occur as the back-end of intensive BPR focus.

In either case it protects the company's interests to perform some preliminary work in this area, if only to provide more confidence in the program planning effort. Indeed, it is beneficial to perform preliminary BPR work prior to or as part of product selection.

This work consists of identifying three areas of future focus:

- the business process areas in which management would like to see some, perhaps dramatic, improvement or in which credible evidence exists that optimal performance is not currently achieved
- the areas where vendor or other product consultants believe that current processes cannot be supported by their unmodified software
- a ranking, in descending order of priority, of the potential areas of focus identified above.

The objective of this exercise is not to proceed to reengineering (though in some cases that could be beneficial, particularly in the case of low hanging fruit); rather, it is to identify, prioritize and quantify the degree of effort, and the costs and benefits associated with reengineering the areas. This intelligence in turn informs the program planning effort; without it the program plan is incomplete and in serious danger of inaccurately estimating required resources, including time.

The priority ranking permits the drawing of a line across the page, separating the items that must be done or should be done and which the client is willing to pay to have done, from those which, taken with others, would be too costly, time-consuming or complex to attempt as part of the implementation. They might be attempted as part of a later phase.

2.3 Prototyping and Testing

The prototyping effort usually represents the lion's share of actual effort involved in a package suite implementation. This process is sometimes called a *Conference Room Pilot*. When performed correctly, most of the actual testing is done as an integral part of prototyping.

The objective of prototyping is to configure package modules to a company's business rules and transaction processing and reporting requirements. It is primarily accomplished by defining transaction and report/query formats, screen options and processing parameters maintained in thousands of tables used by package modules to capture information, transform it and produce outputs from it.

However, in order to define the formats and the rules, extensive analysis of the actual business processes affected and supported by the software is required. Optimally, this in turn requires that Process teams map each process, which is to say graphically depict activities, tasks, decision points, flow of information and hand-offs to other processes, then analyze the picture for weaknesses.

In some implementations where process analysis is neither anticipated as required nor stressed as important by the vendor, prototyping can be a reactive process: internal team members, assisted by product-knowledgeable consultants, will duplicate current process flows and attributes by setting parameter values that determine how the software functions. When the software is found to be unable to support a particular process flow, work will halt until a decision is made as to whether to modify the process or the software. More time will then be required to determine *how* to actually modify the process or the software, then to do it, before configuration work can proceed.

In implementations where process analysis is understood to be an important part of prototyping, the primary effort becomes one of process analysis, aided in real time by product-knowledgeable consultants who know the constraints of the software. Configuration and testing become the back-ends of the prototyping effort.

The second approach is more favorable to situations where management wishes to optimize its processes while the first can save money if the belief proves accurate that little needs to be changed or cannot be supported by the unmodified software.

An important part of either approach is developing policies and procedures for employing the configured software. The Process teams do this; rarely is it done by documentation from the vendor (because the vendor cannot accurately anticipate how a customer will configure the software).

In the event that software modification is required, a specification for the modification is devised jointly between the Process team and the Technology team. The Technology team performs the modification.

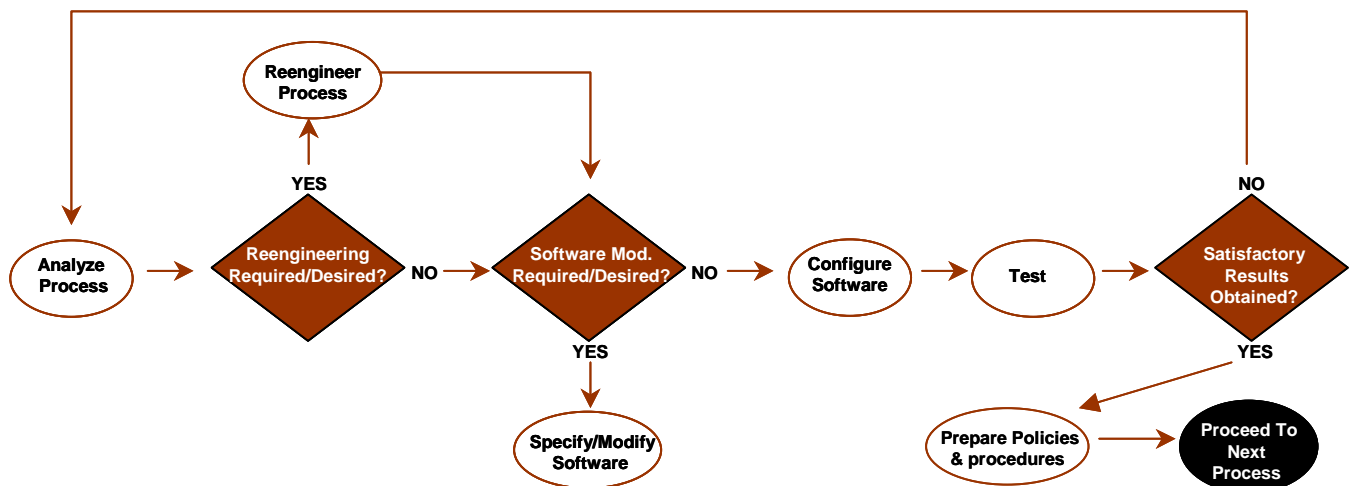
Testing of each part of the prototype is performed immediately subsequent to configuration (or modification), but requires preparation of a testing environment prior to commencement of work by the teams. The testing environment consists of:

- a physical environment supporting an isolated version of the software,
- a *test bed* of test data whose use should produce known results, and
- a set of *test cases* that are intended to represent all situations and conditions (including illogical and other error conditions) relevant to the module or module part undergoing testing.

Process Analysis/Reengineering-Prototyping-Testing is an iterative process. Team members will define process or software changes, configure (or modify the software), then test; if they are not satisfied with results, they may further reengineer, re-configure (or re-modify) and test again. Iterations will continue until satisfactory results are obtained.

When the prototype is complete and its parts have been tested individually, end-to-end (system) testing is performed to determine if hand-offs among modules, module parts and external systems are accomplished accurately and reliably. The entire system is also subjected to a volume test, which determines processing characteristics of the software under high transactional volumes. Finally, when all policies and procedures have been finalized, and selected operational personnel trained, an *Acceptance Test* is performed whose objective is final sign-off of the software by management for roll-out to the organization. The Program Sponsor officially *accepts* the new environment.

The time and effort required to develop reporting and query capabilities in the new environment are often very heavily underestimated in the program planning effort, and care should be taken to allocate sufficient time and human resources to accomplish these activities as per budget.



The prototyping effort, assuming it is structured to emphasize process analysis, can be graphically depicted, as shown above.

2.4 Roll-Out

Roll-out of a fully tested prototype to the organization proceeds according to the Program Plan, and can be done in a *flash-cut* to the entire organization, division by division, nation-by-nation, or even department by department.

It assumes that the Technology teams have designed, programmed and tested any interfaces required to communicate new and legacy environments, and that all historical data conversion has been performed and is ready for use. It should be noted that conversion of historical data is an often-underestimated activity, and that sufficient attention should be given to quantifying the complexity of the task.

It also assumes that all personnel who will use or support the software have been trained in its use or support. This can be accomplished through scheduled training of all personnel by the vendor shortly before the transition date, or by intensive training of internal personnel, who then become trainers of the mass of users.

Parallel concurrent legacy and new environments are increasingly rare today. They were once used extensively to assure that the processing by the new environment was reconcilable with that of the legacy environment, using real data. However, double-input by users to the parallel environments has become very expensive, and such control exercises increasingly are effected as part of standard testing activities.

However, it is common that legacy software is kept “hot” through a transition period which could be one or two months, in the event that the new software should encounter serious problems which would require “stepping back” to the legacy environment until the problems are resolved. This requires that all data input be duplicated, retained and fed (usually automatically, via software) to the legacy environment so that it can remain current in the event it needs to be reactivated.

2.5 Post-Implementation Review

Finally, the Program Plan should include a phase subsequent to actual transition and roll-out, a *Post-Implementation Review*, scheduled perhaps six months to one year beyond initial production use of the new environment. The intent of this phase is to assess whether the objectives set by management for the implementation have been realized; and, if not, to determine why and what might be done.