



BPM Solution Frameworks

Achieving Revolutionary Objectives through Evolutionary Change

By Derek Miers – CEO, BPM Focus

Abstract

Business Process Management offers an alluring future. Increasingly, vendors are enhancing their core offerings with applications targeted at vertical and horizontal segments of industry. However, there is a vast difference in the capabilities of the products available in the market place. This paper explores the value proposition of solution frameworks and offers advice and guidance on their implementation, highlighting many of the challenges, best practices and pitfalls.

About The Author

Derek Miers is a well-known, independent industry analyst, technology strategist, and accomplished author – publishing numerous white papers and product assessments on Business Process Management and process improvement. As Co-Chairman of BPMI.org (the leading BPM standards body committed to the development and introduction of open, royalty free, XML-based standards for business processes) he was instrumental in driving the merger of BPMI.org with the Object Management Group (OMG) and was one of the very early members of the Workflow Management Coalition (WfMC).

He has recently completed the most comprehensive review of BPM environments currently available – the [BPM Suites Report](#) (published via BP Trends). Over the years, he has carried out a wide range of consulting roles, including running hundreds of training courses (in BPM and process modeling techniques), and undertaking detailed technology selection assessments and project risk assessment studies. Other engagements have involved the provision of strategic consulting advice – from facilitating board-level conversations, to supporting BPM project initiatives. Clients have included many of the world's largest and best-known financial services companies (banks, building societies, and insurers), pharmaceutical companies, telecom providers, commercial businesses, product vendors, and governmental organizations.

He can be reached via email at miers@bpmfocus.org by telephone on either +44-20-8742 8500. Other research papers and training course outlines are also available on the Enix web site at www.bpmfocus.org.

Introduction

A rapidly growing number of firms today have correctly identified Business Process Management (BPM) technology as the most effective way of enhancing organizational performance. Whether they want to reduce costs or cycle time, or boost customer satisfaction, they are turning to BPM technology to provide the strategic platform to “operationalize” continuous performance improvement initiatives. Others have discovered the capabilities of BPM while solving technological challenges such as orchestrating the components of a Service Oriented Architecture.

While BPM Suites execute the underlying processes and rules that actively drive the work through the firm, they also provide visibility into the reality of business operations. This enables managers to make better, more informed decisions and usually helps identify the next round of improvement opportunities. When it comes to implementing those changes, the processes and rules used to support the work are easily specialized and improved, dramatically reducing development costs and the time to effect the adjustment.

Moreover, to reduce the time to value, many vendors supply process “templates” to show how their product can support a given business or industry problem. But there is a world of difference between a template and the “solution frameworks” that a few vendors provide. Solution frameworks are customizable applications and building blocks that provide a solid start point for organization specific applications that address common business problems.

This later approach enables firms to get to the point of value a lot quicker, with lower risk. Furthermore, when combined with an iterative improvement methodology, the organization will end up with a much better business fit with its underlying business need. Good BPM solution frameworks provide an opportunity for the business to get quick wins, allowing employees to “smell the coffee” and get behind the change.

But be careful. The term “framework” is widely used, meaning many different things in the BPM arena. Within this paper, we consider the various types of frameworks used in BPM, but focus on the value delivered by “solution frameworks”, highlighting the implementation best practices. The approach taken (by vendors) varies considerably in their attempts to kick start implementation across a number of dimensions:

- The depth of functionality that is included out-of-the-box, reflecting the vendor’s depth of industry expertise (use of data, business rules and process sophistication).
- The development and deployment environment used to support them.
- The amount of further work required to implement effective solutions.
- The extent to which the approach lends itself to a continuous improvement methodology and enterprise wide applicability.

A Running Start

When bringing process-enabled products to market, vendors usually attempt one of two strategies. Either they pursue a “pure-play” approach, with generic technology that is applicable to any business problem, or they tailor their offering to specific applications problems (either vertically or horizontally oriented). Indeed, under the skin, embedded process engines support many niche application products that sell as point solutions. Other vendors have taken a best of both worlds approach in their product architecture, developing a generic BPM platform but then layering on top of that functionality to deliver an effective start point for implementation in a target market.

It is this later approach that we describe as “solution frameworks” (some vendors call them “accelerators”, “templates” or even “pods”). By leveraging the capabilities of the underlying

BPM Suite, they enable the rapid development of new applications, reusing the functionality of existing IT assets embedded in hard-coded enterprise-oriented products (CRM, ERP, etc.) and in-house developed applications.

Solution frameworks provide firms with a running start; enabling an organization to implement a viable solution far more quickly than it would have done if it developed the entire suite of functionality itself. The building blocks that form these applications use the common data structures and business rules appropriate to that vertical industry or generic business area. The core of their value proposition is faster time to market.

When thinking about solution frameworks, a useful analogy is that of building a house. While an architect designed home is unique and offers many opportunities to stamp one's own special character of the environment, in the end it provides a place of shelter and refuge. On the other hand, the vast majority of home dwellers cannot afford this luxury and are happy with a product that is fit for purpose. Prefabricated houses are quick to erect and yet still provide their owners with an opportunity to personalize the environment, but the structure is still relatively fixed. The equivalent of a solution framework would be a set of components that enable the construction of the desired home from well-designed prefabricated parts – walls, floor, roof, plumbing, etc. All of these components are designed to quickly fit together, creating a distinctive environment that delivers the required functionality.

Frameworks come in all shapes and sizes, with varying degrees of efficacy and functionality. They were not created equal. Good solution frameworks provide a solid, well-constructed foundation upon which to build business applications, rather than merely providing cement and an idea. Depending on the approach taken by the vendor, they could be describing a simple set of procedural models (that can be later modified to target a given business problem). Sometimes, these are little more than extended tutorial models. At the other end of the scale, solution frameworks are modular, service-oriented components that provide out-of-the-box application functionality (yet are still customizable to the needs of the business).

However, regardless of how much firms have in common (in one sector or category of application); most organizations consider that the way they do business is unique. Yet on the other hand, it is silly to re-invent the wheel. Therefore, effective solution frameworks must walk a fine line between providing useful functionality and over specifying the solution.

As a result, some solution frameworks are more compelling than others. Apart from the generic processes, they can include the underlying data models, user interface forms, legacy application connectors, and declarative business rules commonly used in the particular business domain. Some encourage and facilitate re-use and specialization in the way they handle versioning, enabling the firm to control pilots and system rollouts, as they fine-tune the layers of the application.

Above all, the best solution frameworks embed significant domain expertise and best practices, cultivated across a large number of enterprises. They are really application toolkits that enable the firm to get up and running very quickly (see the Highmark example below), yet they still support continuous process improvement (the core objective of BPM initiatives). They are focused on either the needs of some vertical industry (such as Retail Banking or Healthcare), or oriented towards generic application problems found in a variety of industries (such as HR, Call Center, Case Handling, Compliance or Procurement). They provide a set of configurable components that accelerate the development of customer specific application solutions. If they have been developed and deployed effectively, then the infrastructure will still be pliable and moldable going forward.

Case Study – Highmark Insurance Group (SideBar)

HM Insurance Company (formerly known as Highmark Life and Casualty Group) provides employer group benefit products. Having undertaken a short business process analysis project beforehand, the group selected the Pegasystems Customer Process Management (CPM) framework for a strategic project to test the efficacy of BPM within one of the Call Centers. From the date of the selection decision, the first implementation took just five weeks (which included initial training).

According to the HLC Project Manager, this was the first project where the technology itself was not the challenge. The implementation team deliberately sought to lower expectations at the outset. They were very clear with the user department that the first release was only a start point. Their primary goal was to ensure a successful project and gain rapid adoption amongst the user community.

From a project scope point of view, they set out with an unambitious target – to keep to the framework implementation as much as possible. Of course, the system required some modification, but this was kept to the absolute minimum in order to avoid “scope creep”. To simplify integration, they had deliberately selected an area of the business with relatively straightforward requirements (just one legacy system). They prototyped the solution to iron out the kinks, with screen mock-ups alongside the process flows.

Before the system had gone in, the high level metric on work turnaround was 2-3 days. Apart from general observations, managers had no hard data to reach this conclusion. After the first month of operation, managers now had data around which they could assess performance. Aggregating all of the service types together, the average turnaround time was now down to 1.3 days. However, when they drilled down into the individual service categories, they found that some took much longer, while the majority of the services were much shorter. Effectively, they had identified those services that needed further attention. The system had given them visibility into the next improvement opportunity. As the Project Manager put it, “Until you actually put in the BPM environment, you don't have the granularity of measurement data to know where to put your effort next.”

Furthermore, they had a technology platform in place upon which they could easily implement those changes. Two days after the system went live; the team had a formal meeting with the end-users and started developing a list of desired enhancements. The first iteration of the development cycle then completed within a week and a half of the initial implementation. One further release later, they had achieved a relatively stable system that delivered real business benefits.

The point is that they kept the initial implementation as simple as possible. While there was a wide range of sophisticated functionality inherent with the Pega platform, the team recognized that the biggest benefit was the provision of an effective measurement capability, enabling them to identify the next round of improvement opportunities. In many cases, these fundamental operational improvement opportunities are simply unknowable up front. Moreover, they carefully considered the impact of the system change on the users. The more complexity that is added (to rules and processes), the longer it is going to take people to learn and become productive following implementation.

A Taxonomy of Frameworks

In the language of management and business, the word “framework” has taken on many different meanings. Indeed, dictionaries provide a plethora of definitions for the term including the following selected examples:

- A frame of reference—a set of ideas, conditions, or assumptions that determine how something will be approached, perceived, or understood.
- An underlying set of ideas—a set of ideas, principles, agreements, or rules that provides the basis or the outline for something that is more fully developed at a later stage.
- In object-oriented systems—a set of classes that embodies an abstract design for solutions to a number of related problems.

Within the context of Business Process Management (BPM),¹ the term framework has also been used to describe:

- A structured methodology for running BPM change projects.²
- A set of documented best practices for IT Service Management (ITIL), or an approach to modeling enterprise architectures (Zachman).
- Methods designed to help organizations in the assessing their maturity, as they evaluate, develop and build an organizational BPM capability.
- Industry Consortia reference models which include:
 - SCOR—the Supply-Chain Operations Reference-model was developed as a cross-industry standard diagnostic tool for supply-chain management. Now extended and adapted with Value Chain oriented VCOR model (from a separate organization).
 - COSO—defines essential enterprise risk management components in the audit and compliance arena.
 - COBIT—an IT governance model designed to bridge the gap between requirements, technical issues and business risks. The approach emphasizes regulatory compliance and helps organizations to increase the value delivered by IT investments.
 - HIPAA and HL7—the Health Insurance Portability and Accountability Act provides national standards for electronic transactions and national identifiers for healthcare providers, health plans, and employers. HL7 is a set of standards for electronic interchange of clinical, financial, and administrative information among health care oriented computer systems. It delivers such functions as security checks, participant identification, availability checks, exchange mechanism negotiations and data exchange structuring.
 - eTOM—enhanced Telecom Operations Map describes the full scope of business processes required by service provider in the telecoms industry.
- A structured set of reference data provided by major vendors, such as IBM's Information FrameWork (IFW) in retail banking and the Insurance Application Architecture (IAA) in the insurance arena. IFW provides "an information architecture blueprint with detailed banking business content that can be applied broadly to many different types of initiatives on an enterprise-wide basis or on a specific project." IAA

¹ A search for Frameworks and BPM will also reveal vendors pushing Business Performance Management (BPfM), relating to sets of metrics for monitoring business performance. This only has a tangential relationship to driving business operations using business processes (the focus of BPM). See the "[BPM Driving Business Performance](#)" on the Enix web site.

² See [The Keys To BPM Project Success](#) and [Developing A Repeatable BPM Capability](#) on the Enix web site. Others include "The Smart Build Methodology" from Pegasystems available on request at www.pega.com

includes business process models for more than 100 specific processes, standard terminology and software code. IAA was recently donated by IBM to the ACORD Insurance Standards body.

- Even a set of integration connectors to attach legacy applications to BPM products. Another vendor describes a BPM framework as a collection of APIs for exposing a BPM Engine as Java, Web services, or XML over HTTP.

Clearly, these are not the thrust of this paper. We are interested in the range of functionality that stretches from “process templates” to “solution frameworks.” They should enable the delivery of some level of application functionality. However, it is important to note that they do this in an entirely different way from traditional applications. Rather than developing a custom application from scratch, or configuring a rigid point solution and then layering on a set of custom extensions (as is the case with many Enterprise Application products), good solution frameworks provide the firm with a way of managing the application as an asset. They provide re-usable building blocks, some of which may be fine-grained chunks of functionality, while others encapsulate broader application features. Think of it a bit like the difference between Lego and Duplo.

Spectrum of Framework Functionality

In their simplest form, some vendors describe their process templates as frameworks. However, such functionality still needs a considerable amount of work to create a realistic user application. The business “context” within which the process will be used is usually outside of the scope. The developer must decide on the data structure that will support the application; integrate with Line of Business (LOB) data sources, create the user interface, build in any related business rules,³ and manage the deployment.

At the next level, a set of user interface forms (and potentially some sample data) complement the process templates. The forms link to the variables defined in process templates. Further development will normally involve integration with LOB data sources. Additional declarative business rules may also be required (usually supported by a third party Business Rules Engine). At this level it is worth checking whether the application supports industry standard data structures (such as the Pega Healthcare Payer Industry framework support for HIPAA, or their Retail and Wholesale Banking Industry Frameworks support of IFW).

Of course, the application itself will normally require adaptation to handle every subtly different use case, each of which will require a subtly different set of processes, business rules, user interface forms and LOB data. Furthermore, as the prevailing business climate evolves, so do the functional requirements for the application. Over time, depending on the approach taken by the vendor, considerable re-work may be required to effect these changes. In most cases, a “copy and paste” approach to development is the result, which does not provide any meaningful way of handling the overall application as an asset. Fragmentation occurs and with it, an exponential increase in complexity (along with the cost of ownership).

³ See “[Rules are from Mars & Processes from Venus](#)” on the Enix web site.

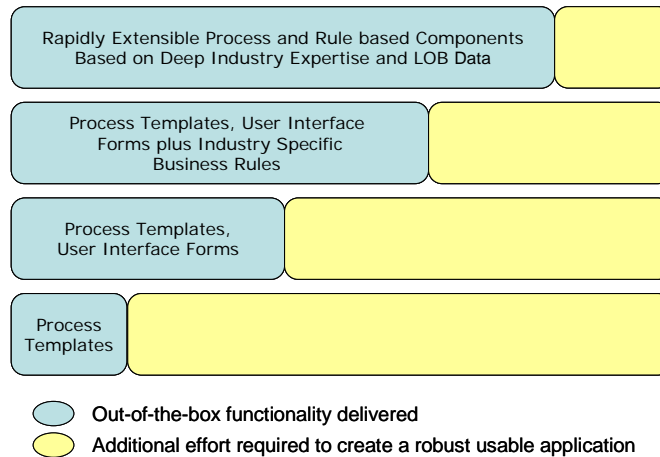


Figure 1 – All application frameworks were not created equal

A number of factors drive the complexity of an application for a given vertical industry: the data structure, the numbers of abstract roles, and the numbers of common declarative rules of the domain.

On the other hand, with an integrated, object-oriented data model underpinning the environment, it becomes possible to “specialize” the application to handle each subtly different use case. Otherwise, there is no way of overriding some of the functionality delivered. Rather than changing the framework itself, it is far better to add a layer on top of that delivers the desired functionality. Along with the notion of specialization is inheritance. Automating the inheritance at runtime makes deployment an order of magnitude easier. At runtime, the system can identify appropriate specialized set of process components and rules that are appropriate to the context of the case in hand.

Dealing With Complexity – Healthcare Example (sidebar)

The HIPAA legislation is transforming the healthcare industry in the United States through regulatory mandate on all healthcare providers and health plans. At the core of the HIPAA regulations are nine key transactions that cover 90% of all transactions between the various parties involved in healthcare; touching payer, member, broker, sponsor and providers alike. Related to that set of transactions is a common data model that enables the exchange of information within these transactions. The legislation also addresses the security and privacy of health data related to individuals.

While the healthcare industry has its own special needs, for information and shared transaction support, it still shares the same requirements for horizontal applications with many other industries (such as Call Centers Support and Case Handling). Clearly, solution frameworks need to interact and work together to provide a seamless environment for the user, combining the building blocks selected by the company from each domain.

Given the existing patchwork of (non-compliant) systems employed by various healthcare providers and health insurers, the potential market for solution frameworks in this area is very large. With a healthcare solution framework already incorporating the core transaction processes, firms can fast track their compliance initiatives by integrating their legacy applications to the complex HIPAA data sets provided by the BPM framework.

Dealing With Change – Finance Example (sidebar)

An emerging and rapidly growing payment type in banking today is ACH (Automated Clearing House). The ACH Network is a reliable and efficient nationwide electronic funds

transfer system that provides the inter-bank clearing of electronic payments. The US Federal Reserve governs its operations. This is the payment network that enables direct debits, bill payment, e-checks, etc. (equivalent to BACS in the UK).

The key change that has occurred recently is that the system now allows paper checks to be converted into ACH payments at the lock-box. Major credit card, utilities, mortgage companies, etc. are converting consumer checks to ACH payments directly, allowing them to recover their receivables faster. As a result, they are improving their liquidity and reducing outstanding receivables. But they are also removing much of the cost of check processing downstream at the consumer's own bank.

While this sounds like good news (for all parties), the problem is that most banks do not have well defined (and automated) processes for handling the exceptions that emerge from ACH transactions. These exceptions might include customer enquiries and disputes, returned and disputed payments from other banks, etc.

To handle that problem, a bank could take the financial adjustment process from their existing retail-banking framework, and then tailor it to handle the ACH payment type. They might then specialize that to cater for each of the 20 different types of returns, developing first the common elements (to handle all ACH scenarios), and then adding specializations that cater for the specific nuances of each scenario. These might include communicating with customers, or performing adjustments, right through to creating the various types of correspondence that are specific to some of those types of exceptions.

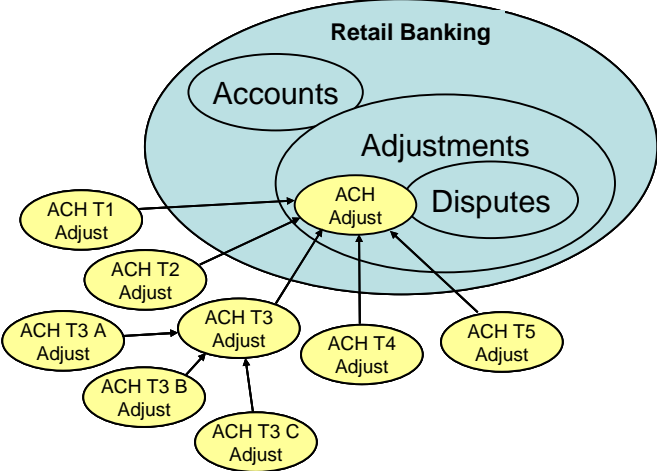


Figure 2 - The Retail Banking Solution Framework from Pegasystems could be specialized and adapted to the nuances of handling ACH exceptions, re-using functionality delivered by the core environment and other solution frameworks

Of course, in this compliance-focused business environment, it necessary to track any changes to processes and rules down to a very granular level, capturing and storing related documents. To handle this, the system itself should provide the change and version management facilities to ensure disclosure transparency, traceability and auditability.

Implementation Pitfalls to Avoid and Best Practices

Regardless of the attractiveness of framework implementations, in common with other BPM initiatives, they run the risk of falling victim to their own potential. Without care, a number of factors can combine to stymie broad adoption and user acceptance:

- Expectations are easily set too high, with business people looking for a silver bullet that will fix all organizational ailments.
- Because business people have become used to the very long development cycles associated with traditional technology approaches, they ask for too much in the initial implementation (forgetting the iterative nature of the methodology). As a result, the implementation team is tempted to increase the scope (“scope creep”), slowing the development lifecycle and threatening overall project success.
- A lack of executive sponsorship will inevitably lead to user resistance and another failed project. Implementing new and improved business processes inescapably implies change (to the way that work is undertaken). Therefore, it is imperative to have the backing of those ultimately responsible for the operations of the business.

Some of these problems derive from the way that vendors have set out to sell their products and educate the market. In their desire to engage potential customers, they set an expectation that their approach will quickly deliver the most dramatic results (big gain, no pain). What they sometimes gloss over is the critical importance of bringing the business along with the initiative.

Business Sponsorship Best Practices

- ✓ *To succeed in the long term, those charged with the initial project need to build momentum, commitment and motivation to ensure that employees will engage on the BPM journey ahead. Ensuring that the first project is successful is the most critical ingredient for ongoing business buy-in. Otherwise the initiative can quickly suffer from a lack of enthusiasm and disillusionment amongst users.*
- ✓ *Ensure that business users take ownership of the BPM program. Do not build a dependency on a system integrator or BPM vendor to run the program. Certainly, vendors and consultants can help, but the business has to own the change program. Creating a BPM Center of Excellence (CoE), building the business acumen and BPM change capability.*
- ✗ *Do not attempt to develop skills and understanding on the capabilities of the selected framework without the support and involvement of the vendor. To get the most out of a framework application, ensure that the vendor has the specialist industry expertise necessary to assist in its initial implementation.*

Think Big – Start Small and Iterate

One key problem to overcome is that business users are normally used to big bang, waterfall style approaches that bring a large amount of new functionality into play at one time. For the best chance of success in the initial project, it is important to educate users to understand the iterative nature of BPM deployments.

Having decided that frameworks represent a viable option, the challenge is to help business managers appreciate that the approach is fundamentally different (from traditional application implementations). It will be necessary to undertake a Gap Analysis on the capabilities of the framework against the perceived needs of the business.

The aim is to identify the essential capabilities that must be included in the initial implementation. For example, does the target system need to integrate with some legacy application that stores the core information today? In which case, it will make sense to build the necessary integration facilities, or to develop substitute functionality (if the legacy system is destined for the scrap heap). Of course, the business will have subtly different

processes and rules from those delivered within the framework, so the challenge is in aligning these two.

While this might sound challenging, it is usually not that difficult. The processes, rules and data contained in the framework will probably represent the common business practices of the domain in question. For instance, in the Call Center of any service organization there is a fundamental set of processes that is common to all organizations. The core need is to validate the caller, capture the intent of that call and then resolve the work items that derive from that intent. Resolving those items is the domain of the business rules and standard operating procedures of the firm. Common elements also include identifying the recipient for each work item, tracking the work, etc. Nevertheless, all of those aspects are driven by the initial verification, capture and resolve phases of the customer interaction.

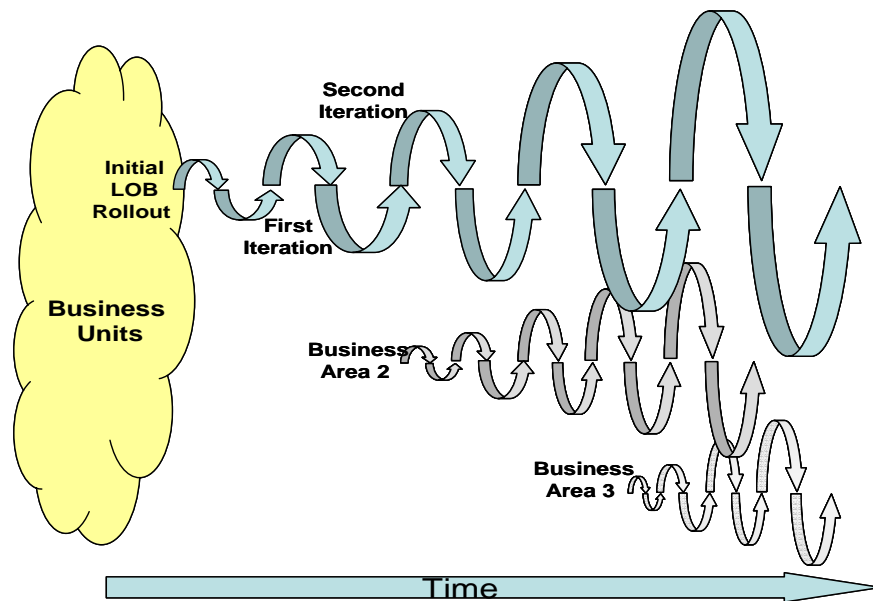


Figure 3 – With a tightly focused initial project, users embrace the change and get behind the initiative, driving wider adoption across the business

While the initial release of the base framework system is likely to be much smaller than a traditional application deployment, in the end, through rapid iteration, the organization will achieve a much better fit to its real, underlying business need. As experience is gained, subsequent business areas achieve more rapid iterations (notice in Figure 3 that the cycles of iteration get tighter and tighter) and will often re-use functionality from the solution framework initially purchased.

Project Planning Best Practices

- ✓ *Keep the high-level vision, but start with a tightly scoped project to solve a discrete problem and then deploy. Build iteratively on that base functionality toward the long-term goal. Initially focus on addressing the immediate need, but keep in mind the big picture, educating the users along the way as to what is possible.*
- ✓ *Rely on iterative development to deliver an ongoing improvement culture.*
- ✓ *As much as is possible, try and use the framework in its “vanilla” form (the out-of-the-box functionality).*

- ✓ *When undertaking the Gap Analysis it is normal to spend time identifying areas of necessary functionality that the framework does not cover out-of-the-box. For the implementation team, this usually involves an operational walk through to understand what the business system does today. Then they will need to compare that with the functionality delivered by the framework. It is probably necessary to involve the solution framework vendor in that exercise.*
- ✓ *As part of the Gap Analysis undertake a design review to make ensure that the solution leverages the capabilities of the components provided. Where the framework delivers areas of automation that are not reflected in current business operations, demonstrate the benefits of this functionality but be prepared to introduce it later. Initially focus on the functionality that the organization deems important.*
- ✓ *Develop a "Road Map" for the users, detailing the short-term vision, medium term vision and long term goals. To some extent this will depend on how visionary they are and what they can see this solution doing. Formally review the Road Map following each new release and add any new functionality as desired to that picture.*
- ✓ *Where possible, show how newly identified areas of functionality will not fit into the scope of the initial project. However, capture the request and add it to the list. Refine the list for the Road Map in terms of a "Must Have" and "Nice To Have".*
- ✓ *Recognize that businesses change rapidly and that requirements change as a result.*
- ✗ *Do not try to do too much, too fast. This will slow the project down and affect the rapid implementation objective of the initial project.*
- ✗ *Avoid adopting a "cut and paste" mentality. Look for the ability to control inheritance in the underlying product to ensure that functionality can be specialized and deployed easily.*
- ✗ *Be aware of the "compliance impact" of the iterative methodology proposed here. In most firms, change management is an important aspect to deploying change. The system should ensure that the appropriate artifacts are gathered and stored appropriately to document the change. It will also need a sophisticated versioning capability to ensure that the processes and rules deployed are consistent with the intended change.*

Work With the People

When implementing any technology, there is only one opportunity to make a good a first impression. Successful adoption of the framework is key to ongoing success. So it is important not overwhelm users and stakeholders with functionality that they do not see as initially important.

No matter the level of automation attempted, for some group of users that will represent change. Change is not comfortable for most people, especially when that change is "being done to them", rather than "driven by them". In a sense, they need to work out for themselves what is needed. There is no point in having a highly sophisticated system full of all sorts of rules and procedural mechanisms, if everyone hates it. Unless users adopt and embrace change, the firm will not get the most out of its technology investment.

Before users can identify the most relevant improvements for their working practices, they need to develop a deep appreciation for what is possible. Something that is impossible with

a one dimensional business requirements document. They need to see it, smell it, touch it – work with it for a while, as some of these improvement options are simply unknowable at the outset.

Remember that those in management (and middle management) have normally been around for some time and have often come up through the ranks. They know how the business works and have the expert knowledge of the business domain. When a BPM team arrives and starts talking about automating their processes and extracting the business rules, then the first thing that they think is that their job is in jeopardy. Resistance is natural under such circumstances.

This creates a big hurdle to overcome. On the other hand, if the users are involved and they quickly see how the system will make their life easier and help them to become more effective, then they will get behind the initiative (rather than resisting it). The overarching methodology of BPM comes into its own as users discover improvement opportunities—as they identify for themselves what is wrong, and how to improve the process.

Management Best Practices

- ✓ *Keep it simple.*
- ✓ *Treat encouraging adoption of the overall BPM approach as a specific goal. Encourage the user population to understand the principle of iterative development and how that gives them control over how their systems will work. In this way, they are more likely to embrace and support change.*
- ✓ *Ensure that help is at hand (on the floor with the users) when the system goes live.*
- ✓ *Showcase successful projects.*
- ✗ *Be cognizant of the fact that this is not just a software project—it is a change management project for the business.*
- ✗ *Avoid carrying out a traditional functional requirements specification and then trying to “back” that into the features available of the framework.*
- ✗ *Do not get hung up on a rigid project methodology – with an iterative approach there is always a certain degree of leeway.*

Measure Performance

The sooner that the system is in use, then the sooner organization can benefit from an iterative development approach. However, iteration first implies discovery of the improvement opportunities. It also implies a regime of checking performance and openly questioning current operating procedures to identify areas where working practices could benefit from further attention. The key point is that it is much easier to identify improvement opportunities once the users can truly see where performance is poor. Indeed, they will push for the improvements themselves (rather than having them imposed upon them). In the Highmark example above, it was only after working with the framework for a month that the users discovered that certain classes of cases took much longer to resolve.

The BPM environment itself will usually provide all necessary infrastructure to capture and track the actions carried out on individual cases of work. However, products vary significantly in the way they support the investigation and understanding of business performance. The “analytics” capabilities of products now enable the creation of so-called

dashboards that provide visibility into current operations, but most leave it up to the user organization to decide what performance means in the context of their business. The designer must decide what information to aggregate, how to monitor and display it and who will have access.

Some frameworks include pre-built mechanisms for performance measurement. In a given domain, the vendor has identified the aspects of performance commonly monitored and has provided user interface screens to support that. For example, in the Pega SmartBPM CPM framework (used by Highmark), the core Pega application is keeping track of each system interaction, right down to mouse clicks and key strokes. In the CPM framework, managers can assess a wide range of metrics that relate to the operation of a typical Call Center. For example, the wrap-up time versus the interaction time, or investigate how long it takes to validate customers. Moreover, the manager can drill down to look at the distribution of cases, or delve into the detailed level of an individual case. In the past, a business analyst with a stopwatch and clipboard undertook that sort of analysis. At Highmark, they can generate a report to assess which workers are taking the longest (and who is quickest). This information could potentially support the employee performance evaluation process (and potentially compensation).

ROI Best Practices

- ✓ *It is a good idea to ensure that the team carries out a fundamental re-assessment of metrics as part of the BPM implementation. Explore how to integrate the business relevant data (such as customer, or product information) with information on cycle-time, resource utilization, etc. When assessing Key Performance Indicators (KPIs), make sure that they support the underlying Key Business Objectives (the firms KBOs). Further, take care to ensure that the measures will reinforce the behavior that the initiative is trying to encourage.*
- ✓ *Formally review the "end-to-end" performance of processes on a regular basis. These end-to-end measures are sometimes called "Capability Measures". They capture the customer's intent such as the time to resolve a case rather than the time spent on a call.*
- ✓ *Rather than arbitrarily setting targets (external to the process), link measures to the reality of the process as it currently performs. Arbitrary targets encourage 'cheating' as people try to beat the system. Indeed, it is a good idea to ensure that at least one target is oriented toward embedding continuous performance improvement within the culture of the organization.*
- ✓ *Recognize that people need to identify improvement opportunities for themselves. An IT driven agenda cannot force business change. The BPM team needs to take on a role of facilitator.*

Conclusion

Solution frameworks provide solid basis for firms to rapidly overhaul their operations and enable better business process management, reducing operational risk. Quite apart from enabling the rapid adoption of industry best practices, the best products also provide the opportunity to differentiate a firm's offerings from those of its competitors.

Moreover, they provide a start point for the rapid iteration necessary to drive BPM adoption across the firm. Implementation timelines of weeks (or just a few months), rather than the six months to a year common with bespoke applications, help to reduce project risk. If approached correctly, they allow employees to embrace the change initiative, as they

discover the improvement opportunities that make sense to them in the context of how they do business.

Nevertheless, beware—framework applications come in many forms, with wildly varying degrees of efficacy. As the Healthcare and Retail Banking examples illustrate, it is important that the framework infrastructure can handle the inherent complexity of the domain and provides the ability to specialize the application to handle subtly different usage scenarios.

Depending on your needs, some products will be more useful than others. Either way, assess the vendors domain expertise and the sophistication reflected in the solution framework. At a minimum, the framework should incorporate the relevant industry data set (such as HIPAA or COSO).

Compare and contrast the development environment that comes with the framework; ensuring that it supports rapid iteration (of process and rules based on the underlying object model). It should also incorporate a refined set of version control techniques that enable deployment into a granular business domain.

In summary, look for the ability to treat the components of the application as assets, controlling and support access to the varying levels of processes, declarative rules and information management. The copy and paste approach espoused by many vendors builds complexity over time and with it, the cost of ownership. If solution frameworks are approached correctly, they provide the opportunity to turn baby steps into continuous improvement and, from there, quantum leaps in business performance.