

BPM, Antidote to IT Jitters

Howard Smith and Peter Fingar

In this true story, a colleague, let's call him Jack to protect the innocent, works in the IT shop of a large organization. He has been involved with Service Management strategy and processes for the past 3-4 years following ITIL (IT Infrastructure Library) principles. Jack explained to us that he has been, "looking at the big picture and trying to implement a tangible set of methods and supporting development tools that, integrated together, would deliver the required functionality to support end-to-end business processes."

Jack is a strategically-focused business-systems-data analyst. He is also pragmatic and has designed and delivered real Service Management systems and solutions. His employer was recently deemed excellent in Applications Maintenance. Like many IT architects, Jack has a Diploma in Business Analysis and has been trained in BPR techniques. He has practiced these techniques on numerous IT projects over the years. Jack has never lost sight of the fact that the objective of excellence in IT process development, is the delivery of excellent business processes.

Recently he attended a seminar we gave, entitled *BPM: The New IT Service*. This is how it was billed:

The IT services market is changing. Companies are demanding further reductions in the cost, both time and resource, of IT development, operations and improvement. At the same time, they expect more than just an IT tower and basic IT services. They are asking the CIO to help them manage their business processes on an ongoing basis, from discovery to execution to optimization. How can these seemingly contradictory requirements be reconciled?

After the seminar we spoke to Jack about his reaction. He noted that, like other movements, BPM would probably face the usual uphill struggle within the organization. Theory and analytic methods are great to a point, but there comes a time when IT has to deliver real tangible solutions. The theory must be put into practice. He was fascinated by the potential of "third wave" BPM tools where "processes are real systems," not just diagrams or analytic methods.

In his work experience, like many of us, Jack found very quickly that the key players in the organization would support his IT strategies and ideas only if he had projects to deliver systems and solutions or if their line managers and directors set specific objectives. He noted that in some cases the required management support started off well, but soon dwindled in the face of day-to-day operations, especially in down turned economic times. Consequently the strategy, theory and good intentions started to lose favor and become fragmented as individual groups returned to their own day-to-day issues requiring considerable attention.

As an example, Jack told us about a Helpdesk manager. He was obsessed with reducing the cost of support calls, with the side effect that he actually drove up the total cost of support processes because of a disconnect with the needs of a new team responsible for creating an Asset Management database. Total cost of process ownership, which was essentially a problem of IT service configuration management, fell between the cracks of different teams working largely in isolation, using stovepipe IT solutions.

Based on Jack's experience, and hearing our message, he concluded that, to be really successful, BPM has to come from the top. Without the support of senior management, he argued, bottom-up change would hit too many internal walls. He noted that, "many technical IT groups do indeed understand the need to establish processes and operate in a structured, joined-up way, but even so, they are often constrained by their management peers' pressing business objectives that can often disrupt the best laid plans of architects and developers." We agreed. If we had a dollar for every point solution implemented by IT by those who should know better, we'd be in Barbados fishing instead of writing articles to articulate the significance of BPM.

He set out to challenge what he perceived as our overly simplistic BPM litany.

Jack told us that, "The two halves of the equation, business process management and the IT that supports it, cannot be separated. Neither can one effectively drive the other. A fine balance must be maintained." He explained by describing two examples that, for him, were key to the "BPM debate."

Project 1. In 1999 Jack designed and delivered a Helpdesk system, a multi-million dollar solution that supported two call-centers employing 1,000 staff members. Jack approached the business requirements with a DSDM (Dynamic Systems Development Method) style, with 3 months of facilitated workshops, prototyping and so on. At the end of this effort he had a good set of business requirements that the key business sponsors were delighted to support. When he tried to translate that into a Helpdesk solution however it fell on its face. The previously chosen Helpdesk toolset, a software package, was just not up to the task. With the toolset decision already cast in stone, he had no choice but to deliver what he could. The business people were disappointed with the result. By contrast, they were pleased to have been involved in crafting the definition of the new Helpdesk system. At that point the project was judged a success. But the translation of those requirements into reality, through the deployment of a packaged solution, customized through hard coding so as to approximate to a subset of their business needs, disappointed them profoundly, especially as their expectations had been raised by the requirements definition project. Their disappointment was confounded by the time and resource they had expended to document their requirements.

Project 2. In 2001 Jack's organization decided to undertake *proper* Asset Management. Based on the Helpdesk experience, Jack's IT group began from the perspective that their existing processes were bound to be poor, no matter how much they were studied, mapped and reengineered, and that they should therefore follow "best practice." Consequently ITIL was used and a packaged Asset Management system was selected because it was the best available tool that supported the ITIL

processes. In effect, in delivering this project, Jack changed the business processes to fit the tool, rather than change the tool to fit an idealized set of requirements that the business had in mind. The success of this approach was hailed by the entire organization. Whereas the Helpdesk project failed, adopting a standard packaged system was seen as a huge win. Look at the benefits. First, the business did not have to spend time worrying about such details such as *requirements*. Second, Jack had little more to do than select a package. That's easy, just turn to any number of IT industry analysts, read their reports and select the package in the upper-right quadrant. That's the one selling best isn't it? That's the one that's been proven as most successful? It must be the right choice. Isn't that so?

Based on his experience in the field, we thought that Jack would immediately resonate with our BPM message. After all, the Helpdesk project was an example of properly engaging with the business in a process design exercise, only to be disappointed in with the reality of packaged software. Based on that experience we felt that Jack would immediately warm to the possibility of using a BPM system to take those designs and put them directly into practice, reusing any useful packaged Helpdesk software that would accelerate the effort. But Jack didn't react as we had expected. Similarly, far from viewing the Asset Management project as a success, we felt it was failure on behalf of Jack's IT organization. In effect they had said, "We give up. We'll actually use the same processes as everyone else (ITIL) and we'll use exactly the same compliant packaged software as well." But Jack also surprised us with his reaction to that assessment.

Jack felt that the two examples gave a different message. He felt that, "regardless of how much we might wish to engineer the processes, manage the processes and so on, we will always be constrained by the vendor applications and platforms that are out there in the market." For him, "the only alternative would be large-scale and largely unmanageable development and maintenance of custom, in-house development, requiring armies of developers and support technicians and all the difficulties in subsequent enhancements and upgrades that bespoke systems bring." Reality hit us.

Jack had attended our seminar, heard the BPM message, and either completely misunderstood it, or did not believe a word of it. He went on to point out that in his organization, "Process streamlining was achieved using off-the-shelf tools and major cost reductions resulted from this strategy." He described this as, "a significant step away from BPM, as the packaged tools available in the market are actually driving requirements, not enabling the deployment of unique business processes." In conclusion he said, "It's a vicious circle I fear, and perhaps we need the application package market leaders to get behind the BPM idea before it will really take off."

We were speechless after hearing Jack's conclusions—had we been talking to a stone wall? The experience taught us, once again, that however many times and in however many ways we explain the distinguishing nature of BPM, even the most experienced and seasoned architects are sometimes unable to see what is under their nose. For as Jack argued, "The two halves of the equation, business process management and the IT software packages that supports it, can not be separated." Yet, once again, that's precisely what a BPM system does.

In Jack's mind, the only alternative to packaged solutions was his preconceived idea of the "large-scale and largely unmanageable development and maintenance of bespoke systems requiring armies of developers and support technicians." Sound familiar? Having grown up in the era of reengineering and ERP systems, that's all Jack knew. His mental model was this: Need a business process? Two possible answers: Develop from scratch or select a package and customize. Despite explanations of BPM given at our seminar, he found it hard to conceive of a third option, even though Slide 6 of our seminar presentation spelled it out in detail. That's only natural, for people cling to that which they already know and understand. His comments indicate deep-seated preconceptions about the development issues surrounding BPM systems that cannot just be swept under the carpet.

As an IT developer, Jack could not believe that things could be different. After all, unlike businesspeople that just pick up and use new tools naturally, Jack knows in detail what a computer is and what a computer system can and cannot do. Jack was extrapolating from his experience of operating systems, application development languages and today's multi-tiered packaged software and its database foundations. No wonder he drew the conclusions he did. His IT-practitioner background blinded him to BPM and will continue to do so until he experiences it in practice. The irony is that, when he does experience a BPM system his response is unlikely to be "hey – how can we put this to work in my organization?" It's much more likely to be technical curiosity, "how does this BPML stuff work, what's Pi Calculus and can I write a process virtual machine as well?" But unable to picture these game-changing technologies in practice, Jack's natural and completely understandable concerns about the nature of software development and IT practices took over. He could not suspend his disbelief even for a moment.

Jack was stuck in an Old IT metaphor, one based on a jaded Business-IT Contract where requirements are *thrown over the wall* into a black-box IT development process, a world business people will never understand and to which they will never be able to effectively contribute. With this traditional IT mental model deep-seated in Jack's psyche, it was 110% certain that he would be skeptical of BPM's ability to avoid the difficulties in subsequent software enhancements and upgrades. Uncontrolled reengineering, without a BPM system capability, *should* give any IT developer the heebie-jeebies.

What never came through in the seminar, and what we now need to go back to Jack to explain, is that we share the same goals. What did he say at the outset? "I've been looking at the big picture and trying to implement a tangible set of methods and supporting development tools that, integrated together, would deliver the required functionality to support end-to-end business processes." Without realizing it, and without using the term "BPM," Jack was trying to build a BPM system for his organization—from scratch.

What Jack really wanted to build, and what BPM systems deliver, is the antidote to conventional software development, an approach that can be summed up in the phrase "too late to change." Too often what is asked for is not necessarily what's wanted. And what the development team understands isn't always what the business users meant. The resulting delivery, though accepted, may disappoint, but by that time it's too late. Too much has already been invested, business benefits have been delayed,

too much disruption would be required to switch tracks and the business ends up with a suboptimal system and reduced business benefits—and unhappy to boot.

To end this unhappy state of affairs, many developers are turning to rapid application development (RAD) techniques that go under many names. The objective of RAD is to reduce risk downstream through quicker feedback and resolution of mismatched expectations. But traditional RAD generates prototypes that are only meant to illustrate the final system. There is still the need to refine and rewrite the prototype for production-capable deployment. Contrast this with a BPM system that allows for the delivery of a solution straight away, missing out one more cause of delay and the potential confusion or reinvention of the requirement. One developer we spoke to in a major company was also well versed in a range of RAD techniques, but now used a BPM system. He had this to say, “In a sense it’s the difference between ‘what you see is what you *will* get’ and ‘what you see is what you *do* get.’” It’s not about just reducing the cost to build, but reducing the cost to adjust, and to do so on a continuous basis. We call this the Cost-to-Risk equation.

Again, what’s a BPM system? It’s a breakthrough method and supporting tools for removing the jitters, that “state of nervous depression or anxiety” that afflicts IT developers over-exposed to business people. Working as they do in the real world and not allowed the luxury of participating in the methodological world of software development, business people exhibit the following traits: changeful, fluid, mobile, mutable, unsettled, unstable, unsteady, variable, inconstant, adaptable, impressionable, plastic, pliant, ever changing, kaleidoscopic, restless, unfixed, inconstant, uncertain, shifty, slippery, capricious, fickle, mercurial, temperamental, ticklish, volatile, busy, inaccessible, chaotic. Without a BPM system, as IT developers, we’d have the heebie-jeebies, too!