

## Managing BPM

December 2005



**Joseph Francis**

[Joseph.Francis@pcor.com](mailto:Joseph.Francis@pcor.com)

[www.bptrends.com](http://www.bptrends.com)

Recently we ran a segment of our design workshops for generating and managing enterprise BPM organizations for a large European company; in defining their standard BPM processes, we discussed with one sub-team their deployment strategies for process projects. I was met with a kind of blank look: "Deployment strategy?" O.K., I dislike that kind of language as much as the next guy (making a mental note to adjust the workshop and the framework wording), so I rephrased it, "How do you usually approach getting defined projects going in order to use the new processes you have designed?" They had no usual approaches, and the idea was somewhat strange to them. "Don't we just hand the specifications to the project teams, and they launch?" Well, that's one kind of deployment strategy. I'd like to spend this article looking at a handful of process deployment strategies, their merits, and their drawbacks.

### Hit and Run

You usually see this type of deployment with large, expensive consulting companies: The client company defines new processes, and then drops them and allows its own business teams to create projects to deploy – I've heard it called cost-saving. Of course, the business teams are not terribly happy having consultants tell them what to do, so you may or may not have any projects or a deployment result. Symptoms of the Hit and Run approach include anger at consultants (and the traditional jokes – "What's a consultant? Someone who takes your watch out of your pocket, tells you the time, and keeps the watch in payment"); dissatisfaction with whatever the approach was used – BPR, BPM, Six-Sigma, you name it; and failure to change process. Process teams who seek to be internal consulting organizations can easily take a "hit and run" attitude. It certainly frees teams up to do more projects quickly. However, they do not last long as organizations. Avoid this approach. I can't tell you how many "hit and run" projects we have seen.

### Pilot and Steer

Personal favorite. For processes and projects that can be divided among individual products, plants, customer segments, product lifecycles, materials, or other types of entities, you can pilot the process change within a small, non-lethal area, with the process team actively engaged in project definition, material, work, and information flow deployment, and carefully monitor the process network performance against requirements. At the right time, the process scales up across organizations, and the BPM team works to steer the rollout successfully. Business process owners feel in control to a high degree, and there's a fair amount of "skin in the game" for the process team. There is no feeling that the process team "runs away." I suggest this approach where you can actually do pilots – but this is not always possible.

### Test and Control

Tricky to execute. Imagine that the BPM team has created a program that substantially redesigns processes that affect materials and procurement, and that are shared among many global process consumers or are shared sales processes affecting a large global customer base, or perhaps are processes controlling global statutory requirements (import/export licenses) that have effectivity dates. In practice, such a program can be very difficult to pilot and steer. However, with a small process team, it may be physically impossible to



## Managing BPM

### by Joseph Francis

December 2005

participate in a global rollout with such a program. (Imagine a team of 5 affecting a key process in 139 countries.) You could look at this, too, as a “hit and run” situation, since they can hand off only to local project teams. They can, however, oversee global testing: They may not be able to work with each individual team, but they can control the results of testing to ensure that the processes that are being turned on work: Test the processes before rollout, test the processes with the users after rollout, and certify that the new processes work before switching off the replaced processes. Participants have substantial skin in the game, so you reduce perception/acceptance problems, but mostly at the central level. Local teams will still grumble.

#### Extended BPM Team

Best accepted, but heavier investment. For process programs that do not need to be centrally directed (BPM programs not focused on shared process resources), I have preferred to train a local team in BPM standards and methodology that relate to their problem, and to allocate a single BPM resource to coach and manage the extended team for deployment. The drawback is that the overall program is moderately slower. The benefit is that you may now have networks of parallel process teams working to re-engineer local processes, that can also coordinate and work with global resources on global programs in a defined, modular way – a sort of hub-and-spoke. Standard process reference frameworks like SCOR/DCOR are a must.

#### The Borg

Heaviest investment of all – a must for risky programs where you are not worried about friends. To quote the wiki<sup>1</sup> on Borg (short for Cyborg – however, in several European languages it means “fortress”):

They are known both within and beyond Star Trek fan-dom for their relentless pursuit of that which they wish to assimilate or acquire, their rapid adaptability to almost any defense, and their ability to continue functionally after what may seem a devastating or even fatal blow, as if nothing at all had happened, and, as such, have become a powerful symbol in popular culture for any seemingly unstoppable force, against which “resistance is futile.”

With a huge process change, you temporarily create a very large rollout team, which fan-out globally to coordinate the introduction of a new processes. It requires enough people to handle each change event at the local level, and to insure substantial communication with each other. Once the process team mobilizes this group, the change can happen very fast, but, obviously, for such a large resource base, funding demands can be significant. Other terms we used to use for this were “Deathstar,” or the sarcastic “We’re from corporate; we’re here to help you.”

#### Shelf Deployment

Lethal. Your team does a large BPM program; the sponsor, stakeholders, and steering committee like the result, which is put “on the shelf.” I use the term, “shelf,” in reference to software that is purchased in companies for marketing purposes – “shelfware” – and then not actually deployed or used. In some cases,



**Managing BPM****by Joseph Francis**

December 2005

a BPM program sponsor may support a program for the purpose of activating a lot of work in a subject that they have no intention of using; they only want to ensure that there is visible activity (“marketing”). In one situation, my team found that it was one of eighteen different teams working on the same problem. I can understand it if you are working to see what the best result will be. But eighteen different teams? My teams have tried and tried to profile such situations before a program gets launched, but they are difficult to forecast. The “Shelf Deployment” is lethal because the BPM team is the face of the sponsor to the larger internal world, and everyone is aware when all the work and meetings attended lead to no result. The BPM team here kills its own metrics – numbers of processes examined or modified, versus benefit; the numbers of BPM people working versus numbers of major processes changed; and so on.

So, in summary, this is a sampler of deployment strategies. The strategies that are valuable have key features that lead us to select them, such as whether the deployment can be split into pieces, whether it must be done globally and quickly, and so on. Certain deployments that have low interaction and “skin in the game” should be consistently avoided – such as “hit and run” or “shelf deployment” – because they provide poor value to the process owner and will reflect badly on the central BPM team. I would love to hear other examples from readers of their deployment strategies.

