

# If We've Got SaaS, Do We Need BPO?

A look back at the days prior to software-as-a-service reveals a costly and inflexible time in which HR leaders had very few choices for service delivery. By Naomi Bloom

**H**R leaders and HRM specialists have a big stake in using technology to help them achieve the business outcomes (not just the HR department's outcomes) needed by their organizations. They want to improve the quality of their hire decisions, engage the workforce in productive behaviors, enhance the capabilities of that workforce, manage key employee segments on a global basis, reduce or hold steady their overall compensation and benefits budgets while achieving the same or better business results, increasing workforce agility, and fostering collaborative learning and OJT mentoring. The list is long and very challenging.

But there's little value to HRM applications software (including those used by HRO providers) if it limits the creativity of strategic business processes and rules, takes forever to implement, or calcifies once implemented. And that's before we get to the question of how that software impacts the total cost of service delivery (TCSD), let alone the total cost of delivering HRM business outcomes (TCBO).

Whether as part of an ERP, a standalone core HRMS, or the often-needed add-ons from best-of-breed vendors, a wide range of options in product functionality was available through these channels before the late 1990s. However, most packaged applications were obtained via a perpetual license (with annual maintenance payments due in perpetuity) implemented on premise (with or without systems integrators) and designed to serve a single enterprise. And while some outsourcing providers—notably long-established payroll providers—hosted their payroll software (but not their HR applications) and made them available by subscription, this was the very obvious exception to the licensed, on-premise rule.

The **business model** was for those package vendors to sustain fairly long and complex pursuits (interesting that the sales process for big-ticket software, consulting, systems integration, and now outsourcing advisory is described in terminology taken from the hunt or war) in search of mega-buck, CAPEX up-front perpetual license fees and the recurring revenue of annual maintenance fees. Vendors got their money up front, whether or not the software was ever implemented fully and regardless of the extent to which the software was utilized. When you combined that license fee with 20-plus-percent, year-

over-year maintenance payments, the profit margin on this business was huge once the software vendor had recovered the costs to deliver that first release. And those margins stayed huge unless the vendor had to invest in an architectural or object model leap while sales of older/dead-end products declined.

The **deployment option** for those packages was in the client's own on-premise data center or via a third party's rent-a-center, with legions of their own staff, vendor staff, or staff paid for on a time and materials basis from system integrators doing the implementation work. Because periodic, expensive, and difficult upgrades had to be applied, many of those third parties never left. And because their contracts were invariably time and material, they were happy to customize packages to customer demands. Why push unpleasant organizational change?

The **architecture** of these licensed packages was designed to support one enterprise (or just one part of an enterprise bound by geography or line of business) running on one instance of the software with one set of databases specific to that enterprise. This architecture was known as single tenancy, but no one ever called it that because why would you want anything else? There was a lot of complaining about the license and implementation time and costs, the infrequent and very painful upgrades, and the tendency for these packages to constrain through calcification what they were supposed to be enabling, but no one was challenging their basic, single tenancy architecture. However, those same early and quite successful payroll service bureau pioneers ran their businesses on payroll software designed to serve many customers in a single instance.

Selecting, licensing, implementing, and building-out from these ERP/HRMSs was expensive, time consuming, and risky, but these undertakings were not very hard to understand from a business model, architecture, or deployment options perspective. And there was considerable comfort in knowing that you controlled (really your IT organization controlled) the data center where your data resided and which operated your software. This was all before software-as-a-service came along.

In next month's column, I'll discuss how and why we got to SaaS, followed by the pivotal question: If we've got SaaS, do we need BPO? For those of you who can't wait, the answer is: It depends. **HRO**



Naomi Lee Bloom, Managing Partner, Bloom & Wallace, can be reached at 239-454-7305 or naombloom@mindspring.com. You can also follow her on Twitter @InFullBloomUS.